The Behavior of TCP and Its Extensions in Space

by

Ruhai Wang and Stephen Horan

NMSU-ECE-01-008

NMSU-ECE-01-008

The Behavior of TCP and Its Extensions in Space

Ruhai Wang and Stephen Horan

Manuel Lujan Space Tele-Engineering Program
New Mexico State University
Las Cruces, NM

Prepared for

NASA Goddard Space Flight Center
Greenbelt, MD

under Grant NAG5-9323

July 27, 2001

Klipsch School of Electrical and Computer Engineering
New Mexico State University
Box 30001, MSC 3-O
Las Cruces, NM 88003-8001

THE BEHAVIOR OF TCP AND ITS EXTENSIONS IN SPACE

BY

RUHAI WANG

A Dissertation submitted to the Graduate School

in partial fulfillment of the requirements

for the degree

Doctor of Philosophy, Engineering

Specialization in: Electrical Engineering

New Mexico State University

Las Cruces, New Mexico

August 2001

"The Behavior of TCP and Its Extensions in Space," a dissertation prepared by Ruhai Wang in partial fulfillment of the requirements for the degree, Doctor of Philosophy, has been approved and accepted by the following:

Timothy J. Pettibone
Dean of the Graduate School

Stephen Horan
Chair of the Examining Committee

Date

Committee in charge:

Dr. Stephen Horan, Chair

Dr. Dennis L. Clason

Dr. Ray Lyman

Dr. John P. Mullen

Dr. Javin Taylor

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Stephen Horan for his great advice and consistent support through my research during my graduate studies at New Mexico State University. I wish to extend my thanks to my committee members for their advice, suggestions, comments to prepare this dissertation: Dr. Javin Taylor, Dr. Ray Lyman, Dr. John Mullen and Dr. Dennis Clason and particularly to Dr. Dennis Clason for his invaluable help in statistical analysis of the experimental data. In addition, the assistance of Patrick Feighery, Keith Scott and Bob Durst of MITRE to configure and understand the SCPS protocols is gratefully acknowledged. The assistance of David Beering and Greg Romaniak of Infinite Global Infrastructure (IGI) and the assistance of Mike Rupar of Naval Research Laboratory (NRL) to provide satellite link is also acknowledged.

Certainly, I would especially like to thank NASA for their support under the Grant NAG5-9323 and the experts of NASA Goddard Space Flight Center, and NASA Glenn Research Center and JPL who are interested in our study.

VITA

1972	Born at Yougdeng, China
1997	M.S. in Telecommunications with Honors from Roosevelt University, Chicago, Illinois
1998-2001	Research Assistant, Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, New Mexico

PROFESSIONAL AND HONORARY SOCIETY

Who's Who among Students in American Universities

PUBLICATIONS

Stephen Horan and Ruhai Wang, "Internet-Type Protocol Testing in a Simulated Small Satellite Environment," *Proc. IEEE Conference on Aerospace and Electronic Systems*, March 2001.

Stephen Horan and Ruhai Wang, "Effects of Networking Configurations on Channel Throughput in Space," *Proc. 14th AIAA Conference on Small Satellites*, August 2000.

Stephen Horan and Ruhai Wang, "Comparison of File Transfer Using SCPS FP and TCP/IP FTP over a Simulated Satellite Channel," *Proc. International Telemetry Conference*, Vol. 35, October 1999.

ABSTRACT

THE BEHAVIOR OF TCP AND ITS EXTENSIONS IN SPACE

BY

RUHAI WANG

Doctor of Philosophy, Engineering

Specialization in Electrical Engineering

New Mexico State University

Las Cruces, New Mexico, 2001

Dr. Stephen Horan, Chair

The performance of Transmission Control Protocol (TCP) in space has been examined from the observations of simulation and experimental tests for several years at National Aeronautics and Space Administration (NASA), Department of Defense (DoD) and universities. At New Mexico State University (NMSU), we have been concentrating on studying the performance of two protocol suites: the file transfer protocol (ftp) running over Transmission Control Protocol/Internet Protocol (TCP/IP) stack and the file protocol (fp) running over the Space Communications Protocol

Standards (SCPS)-Transport Protocol (TP) developed under the Consultative Committee for Space Data Systems (CCSDS) standards process. SCPS-TP is considered to be TCP's extensions for space communications.

This dissertation experimentally studies the behavior of TCP and SCPS-TP by running the protocol suites over both the Space-to-Ground Link Simulator (SGLS) test-bed and realistic satellite link. The study concentrates on comparing protocol behavior by plotting the averaged file transfer times for different experimental configurations and analyzing them using Statistical Analysis System (SAS) based procedures. The effects of different link delays and various Bit-Error-Rates (BERs) on each protocol performance are also studied and linear regression models are built for experiments over SGLS test-bed to reflect the relationships between the file transfer time and various transmission conditions.

The results from the test-bed show that protocols do not show performance difference with a very small file (≤ 1Kbytes) for all configurations and protocols perform differently with the increase of file size, BER and link delay for both symmetric and asymmetric channel rates. Under this condition, Vegas congestion control based SCPS-TP protocol (SCPS-Vegas) performs superiorly than Van Jacobson (VJ) congestion control based TCP and SCPS-VJ protocols. We also conclude from the experiments over test-bed that the factors of file size, BER and link delay and all their interactions contribute significantly to protocol performance. The results over the satellite link show that all protocols don't have significant performance difference for 115,200 bps:115,200 bps channel rate and protocols show

significant difference for all large files with higher channel rates. The experiments with error free and 120 ms delay also show that SCPS-VJ shows the highest throughput in all cases and SCPS-Vegas shows the slowest throughput. Linearly correlated file transfer time relationship between the test-bed and satellite link shows that SGLS test-bed works validly and it can be used to predict the relative performance of protocols over realistic satellite link.

Additional work with higher BERs and longer delays over satellite link needs to be done to study the effects of the BER and delay to the protocol performance over satellite when satellite link is configured properly. This might also provide us data to compare the protocol performance over test-bed and satellite link for configurations with high BERs and longer delays to verify the above results.

TABLE OF CONTENTS

LIST OF TABLES xiii

LIST OF FIGURES xvi

LIST OF ACRONYMS xxi

1 INTRODUCTION 1

2 TCP PROTOCOL AND ITS EXTENSIONS FOR SPACE 5

2.1 TCP Protocols 5

2.1.1 TCP Tahoe 7

2.1.2 TCP Reno 10

2.1.3 TCP SACK 12

2.1.4 TCP Vegas 14

2.2 TCP Problems in Space 19

2.2.1 High Transmission Errors 19

2.2.2 Asymmetric Channels 19

2.2.3 Overhead of TCP Protocol Header 20

2.2.4 Large Bandwidth-Delay Product 20

2.2.5 Intermittent Connectivity and Variable RTT 21

2.3 TCP Extensions for Space Communications 21

2.3.1 Overcoming Losses from Different Sources 22

2.3.1.1 Congestion-Induced Loss 22

2.3.1.2 Corruption-Induced Loss 22

2.3.1.3 Link Outage Loss 23

2.3.2 Coping with Asymmetric Channels 23
2.3.3 Relieving Bandwidth Constraints 23
2.3.3.1 SCPS-TP Header Compression 23
2.3.3.2 SCPS-TP SNACK 24
2.3.3.3 Other Techniques for Coping with Errors 25
3 SGLS TEST-BED AND EXPERIMENT METHODOLOGIES 27
3.1 Tests over Simulated Test-bed 27
3.1.1 Channel Simulator 27
3.1.2 Experiment Tools Used in NMSU Test-bed 29
3.1.2.1 Expect Scripts to Automate Tests 30
3.1.2.2 Tcpdump, Tcptrace and Xplot 30
3.1.3 Protocol Layers and Configurations 31
3.1.3.1 Protocol Software Entities 31
3.1.3.2 SCPS and TCP/IP Protocol Layers 32
3.1.3.3 Protocol Configurations 33
3.1.4 Experiment Assumptions 41
3.1.4.1 Number of Test Runs, File Order and File Size 42
3.1.4.2 SGLS Test Configuration 44
3.1.5 Experiment Procedures 44
3.1.5.1 Test Method 44
3.1.5.2 Data Collection 45
3.1.5.3 Analysis Techniques 45

3.2 Experimental Work 48

3.2.1 Experiments over SGLS Test-bed 48

3.2.2 Experiments over Satellite Link 55

4 BEHAVIOR OF TCP/IP AND SCPS OVER THE SGLS TEST-BED 61

4.1 Comparing the Performance of TCP and SCPS 63

4.1.1 Comparison with Symmetric Channel Rate of 115,200 bps:115,200 bps 63

4.1.1.1 BER=0 63

4.1.1.2 BER=1E-6 69

4.1.1.3 BER=1E-5 71

4.1.2 Comparison with Asymmetric Channel Rate of 115,200 bps:2400 bps 76

4.1.2.1 BER=0 76

4.1.2.2 BER=1E-6 80

4.1.2.3 BER=1E-5 80

4.2 Delay Effects on Protocol Performance 87

4.2.1 Delay Effects on TCP/IP Performance 87

4.2.2 Delay Effects on SCPS-VJ Performance 91

4.2.3 Delay Effects on SCPS-Vegas Performance 99

4.3 Bit-Error-Rate (BER) Effects on Protocol Performance 106

4.3.1 Bit-Error-Rate Effects on TCP/IP 107

4.3.1.1 Symmetric Channel Rate 107

4.3.1.2 Asymmetric Channel Rate 110

4.3.2 Bit-Error-Rate Effects on SCPS-VJ 115

4.3.2.1 Symmetric Channel Rate 115

4.3.2.2 Asymmetric Channel Rate 118

4.3.3 Bit-Error-Rate Effects on SCPS-Vegas 121

4.3.3.1 Symmetric Channel Rate 121

4.3.3.2 Asymmetric Channel Rate 124

5 BEHAVIOR OF TCP/IP AND SCPS OVER A SATELLITE LINK 131

5.1 Comparing the Performance of TCP/IP and SCPS 133

5.2 Comparing Protocol Performance over SGLS Test-bed and Satellite Link 150

6 SUMMARY, CONCLUSIONS AND FUTURE WORK 161

6.1 Protocols and Experimental Methodologies 162

6.2 Study of Protocols over SGLS Test-bed 163

6.2.1 TCP/IP and SCPS Performance Comparison 163

6.2.2 Delay Effects on Protocol Performance 164

6.2.3 BER Effects on Protocol Performance 165

6.3 Study of Protocol Performance over Satellite Link 166

6.3.1 Comparing Protocol Performance of TCP/IP and SCPS 167

6.3.2 Comparing Protocol Performance over SGLS Test-bed and Satellite Link 168

6.4 Future Work 169

APPENDICES

A. MEANS AND STANDARD DEVIATIONS OF TCP/IP FILE TIMES OVER THE SGLS TEST-BED 172

B. MEANS AND STANDARD DEVIATIONS OF SCPS-VJ FILE TIMES OVER THE SGLS TEST-BED 174

C. MEANS AND STANDARD DEVIATIONS OF SCPS-VEGAS FILE TIMES OVER THE SGLS TEST-BED 176

D. MEANS AND STANDARD DEVIATIONS OF PROTOCOL FILE TIMES OVER SATELLITE LINK 178

REFERENCES 180

LIST OF TABLES

2.1	Summary of TCP variants	6
3.1	Expected qualitative effects of space channel conditions on protocols	52
4.1	Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms	65
4.2	Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=1280 ms	66
4.3	Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=1E-6 and Delay=120 ms	72
4.4	Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=1E-6 and Delay=1280 ms	73
4.5	Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=1E-5 and Delay=120 ms	74
4.6	Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=1E-5 and Delay=1280 ms	75
4.7	Comparison of means for channel rate of 115,200 bps:2400 bps, BER=0 and Delay=120 ms	78
4.8	Comparison of means for channel rate of 115,200 bps:2400 bps, BER=0 and Delay=1280 ms	79
4.9	Comparison of means for channel rate of 115,200 bps:2400 bps, BER=1E-6 and Delay=120 ms	83
4.10	Comparison of means for channel rate of 115,200 bps:2400 bps, BER=1E-6 and Delay=1280 ms	84
4.11	Comparison of means for channel rate of 115,200 bps:2400 bps, BER=1E-5 and Delay=120 ms	85
4.12	Comparison of means for channel rate of 115,200 bps:2400 bps, BER=1E-5 and Delay=1280 ms	86

4.13 TCP/IP comparison of means for two delays with channel rate of 115,200 bps:115,200 bps and error rate of 0, 1E-6 and 1E-5 89

4.14 TCP/IP comparison of means for two delays with channel rate of 115,200 bps:2400 bps and error rate of 0, 1E-6 and 1E-5 93

4.15 SCPS-VJ comparison of means for two delays with channel rate of 115,200 bps:115,200 bps and error rate of 0, 1E-6 and 1E-5 96

4.16 SCPS-VJ comparison of means for two delays with channel rate of 115,200 bps:2400 bps and error rate of 0, 1E-6 and 1E-5..... 98

4.17 SCPS-Vegas comparison of means for two delays with channel rate of 115,200 bps:115,200 bps and error rate of 0, 1E-6 and 1E-5..... 101

4.18 SCPS-Vegas comparison of means for two delays with channel rate of 115,200 bps:2400 bps and error rate of 0, 1E-6 and 1E-5..... 103

4.19 TCP/IP comparison of means for different BERs over symmetric channel with a delay of 120 ms..... 109

4.20 TCP/IP comparison of means for different BERs over symmetric channel with a delay of 1280 ms..... 111

4.21 TCP/IP comparison of means for different BERs over asymmetric channel with a delay of 120 ms..... 113

4.22 TCP/IP comparison of means for different BERs over asymmetric channel with a delay of 1280 ms..... 114

4.23 SCPS-VJ comparison of means for different BERs over symmetric channel with a delay of 120 ms..... 116

4.24 SCPS-VJ comparison of means for different BERs over symmetric channel with a delay of 1280 ms..... 117

4.25 SCPS-VJ comparison of means for different BERs over asymmetric channel with a delay of 120 ms..... 119

4.26 SCPS-VJ comparison of means for different BERs over asymmetric channel with a delay of 1280 ms..... 120

4.27 SCPS-Vegas comparison of means for different BERs over symmetric channel with a delay of 120 ms..... 122

4.28 SCPS-Vegas comparison of means for different BERs over symmetric channel with a delay of 1280 ms 123

4.29 SCPS-Vegas comparison of means for different BERs over asymmetric channel with a delay of 120 ms 125

4.30 SCPS-Vegas comparison of means for different BERs over asymmetric channel with a delay of 1280 ms 127

4.31 Effects of space channel conditions on each protocol performance 130

5.1 Satellite comparison of means for channel rate of 115,200 bps: 115,200 bps, BER=0 and Delay=120 ms 134

5.2 Satellite comparison of means for channel rate of 4 Mbps:4 Mbps, BER=0 and Delay=120 ms 136

5.3 Satellite comparison of means for channel rate of 4 Mbps:57,600 bps, BER=0 and Delay=120 ms 137

5.4 A portion of transmission data of TCP/IP, SCPS-VJ and SCPS-Vegas for transmitting 1000 Kbytes data with rate of 4 Mbps:57,600 bps, BER=0 and Delay=120 ms 139

5.5 Comparison of means for test source for TCP/IP with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms 152

5.6 Comparison of means for test source for SCPS-VJ with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms 153

5.7 Comparison of means for test source for SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms 154

LIST OF FIGURES

2.1	Widely used TCP variants	18
3.1	A typical satellite link model	27
3.2	SGLS hardware configuration	29
3.3	Protocol layers of SCPS and software entities	33
3.4	Flow of packets over PPP for both protocols	35
3.5	Outline showing test factors and different levels of each factor for experiment over SGLS test-bed	48
3.6	Outline showing test factors and different levels of each factor for experiment over satellite link	55
4.1	File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms	65
4.2	File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=1280 ms	66
4.3	File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=1E-6 and Delay=120 ms	72
4.4	File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=1E-6 and Delay=1280 ms	73
4.5	File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=1E-5 and Delay=120 ms	74
4.6	File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=1E-5 and Delay=1280 ms	75

4.7 File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:2400 bps, BER=0 and Delay=120 ms 78

4.8 File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:2400 bps, BER=0 and Delay=1280 ms 79

4.9 File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:2400 bps, BER=1E-6 and Delay=120 ms 83

4.10 File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:2400 bps, BER=1E-6 and Delay=1280 ms 84

4.11 File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:2400 bps, BER=1E-5 and Delay=120 ms 85

4.12 File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:2400 bps, BER=1E-5 and Delay=1280 ms 86

4.13 TCP/IP performance over symmetric error free channel 88

4.14 TCP/IP performance over symmetric channel with error rate 1E-6 88

4.15 TCP/IP performance over symmetric channel with error rate 1E-5 88

4.16 TCP/IP performance over asymmetric error free channel 92

4.17 TCP/IP performance over asymmetric channel with error rate 1E-6 ... 92

4.18 TCP/IP performance over asymmetric channel with error rate 1E-5 ... 92

4.19 SCPS-VJ performance over symmetric error free channel 95

4.20 SCPS-VJ performance over symmetric channel with error rate 1E-6 95

4.21 SCPS-VJ performance over symmetric channel with error rate 1E-5 95

4.22 SCPS-VJ performance over asymmetric error free channel 97

4.23 SCPS-VJ performance over asymmetric channel with error rate 1E-6 97

4.24 SCPS-VJ performance over asymmetric channel with error rate 1E-5 97

4.25 SCPS-Vegas performance over symmetric error free channel 100

4.26 SCPS-Vegas performance over symmetric channel with error rate 1E-6 100

4.27 SCPS-Vegas performance over symmetric channel with error rate 1E-5 100

4.28 SCPS-Vegas performance over asymmetric error free channel 102

4.29 SCPS-Vegas performance over asymmetric channel with error rate 1E-6 102

4.30 SCPS-Vegas performance over asymmetric channel with error rate 1E-5 102

4.31 TCP/IP performance for different BERs over symmetric channel with a delay of 120 ms 109

4.32 TCP/IP performance for different BERs over symmetric channel with a delay of 1280 ms 111

4.33 TCP/IP performance for different BERs over asymmetric channel with a delay of 120 ms 111

4.34 TCP/IP performance for different BERs over asymmetric channel with a delay of 1280 ms 114

4.35 SCPS-VJ performance for different BERs over symmetric channel with a delay of 120 ms 116

4.36 SCPS-VJ performance for different BERs over symmetric channel with a delay of 1280 ms 117

4.37 SCPS-VJ performance for different BERs over asymmetric channel with a delay of 120 ms 119

4.38 SCPS-VJ performance for different BERs over asymmetric channel with a delay of 1280 ms 120

4.39 SCPS-Vegas performance for different BERs over symmetric channel with a delay of 120 ms..... 122

4.40 SCPS-Vegas performance for different BERs over symmetric channel with a delay of 1280 ms..... 123

4.41 SCPS-Vegas performance for different BERs over asymmetric channel with a delay of 120 ms..... 125

4.42 SCPS-Vegas performance for different BERs over asymmetric channel with a delay of 1280 ms..... 127

5.1 Test factors and different levels of each factor for experiments over satellite link..... 132

5.2 Satellite time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms..... 134

5.3 Satellite time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 4 Mbps:4 Mbps, BER=0 and Delay=120 ms..... 136

5.4 Satellite time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 4 Mbps:57,600 bps, BER=0 and Delay=120 ms 137

5.5 Time sequence numbers with respect to file transfer time for three protocols transmitting 1000 Kbytes data with rate 4 Mbps: 57,600 bps, BER=0 and Delay=120 ms..... 140

5.6 Enlarged view of the end portion of the time sequence plots in Figure 5.5 140

5.7 Time sequence numbers of TCP/IP transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms 141

5.8 Time sequence numbers of SCPS-VJ for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms..... 141

5.9 Time sequence numbers of SCPS-Vegas for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms 142

5.10 Enlarged view of time sequence graph of TCP/IP in Figure 5.7 142

5.11 Enlarged view of time sequence graph of SCPS-VJ in Figure 5.8 143

5.12 Enlarged view of time sequence graph of SCPS-Vegas in Figure 5.9 143

5.13 Congestion window graph of three protocols for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms 144

5.14 RTT graph of TCP/IP for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms 145

5.15 RTT graph of SCPS-VJ for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms 146

5.16 RTT graph of SCPS-Vegas for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms 146

5.17 Throughput graph of three protocols for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms..... 147

5.18 Test source comparison for TCP/IP with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms 152

5.19 Test source comparison for SCPS-VJ with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms 153

5.20 Test source comparison for SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms 154

5.21 File transfer time relationship between SGLS test-bed and satellite link for all three protocols 156

5.22 TCP/IP file transfer time relationship between SGLS test-bed and satellite link 157

5.23 SCPS-VJ file transfer time relationship between SGLS test-bed and satellite link 158

5.24 SCPS-Vegas file transfer time relationship between SGLS test-bed and satellite link 158

LIST OF ACRONYMS

ACK	Acknowledgement
ACTS	Advanced Communications Technology Satellite
ANOVA	Analysis of Variance
BER	Bit-Error-Rate
BDP	Bandwidth-Delay Product
bps	Bits per second
CCSDS	Consultative Committee for Space Data Systems
d.f.	degrees of freedom
DoD	Department of Defense
FACK	Forward ACK
fp	File Protocol
ftp	File Transfer Protocol
GEO	Geostationary Orbit
GLM	General Linear Model
HSD	Tukey's Honestly Significant Difference
ICMP	Internet Control Message Protocol
IGI	Infinite Global Infrastructure
IGMP	Internet Group Management Protocol
Kbyte	1000 bytes
LEO	Low Earth Orbit

LFN	Long, Fat Pipe Network
Mbyte	1,000,000 bytes
MHz	1,000,000 Hertz
ms	millisecond
NASA	National Aeronautics and Space Administration
NRL	Naval Research Lab
NMSU	New Mexico State University
PC	Personal Computer (Intel/Windows based configuration)
PPP	Point-to-Point Protocol
RI	Reference Implementation
RTO	Retransmission Time Out
RTT	Round Trip Time
SACK	Selective ACK
SAS	Statistical Analysis System
SCPS	Space Communications Protocol Standard
SCPS-FP	SCPS File Protocol
SCPS-NP	SCSP Network Protocol
SCPS-SP	SCPS Security Protocol
SCPS-TP	SCPS Transport Protocol
SGLS	Space-to-Ground Link Simulator
SNACK	Selective Negative ACK
TCP/IP	Transmission Control Protocol/Internet Protocol

TDRS	Tracking and Data Relay Satellite
UDP	User Datagram Protocol
VJ	Van Jacobson

1 INTRODUCTION

The use of Internet-type protocols for space communications is no longer considered a "new" topic of investigation. Research on the subject of Transmission Control Protocol (TCP) in space has been conducted at National Aeronautics and Space Administration (NASA), Department of Defense (DoD), contractor, and university facilities for several years now. Much of the research has examined the performance of TCP in space from observations of simulation and experimental test-bed results [1], [2]. In [1], the study results of a comprehensive performance of TCP protocol improvements in the satellite environment over error free links under the congestion loss option were presented. The Selective Acknowledgements (SACK) [3] option of TCP was also examined by comparing with traditional TCP implementations. In [2], sets of experiments were conducted using the Advanced Communications Technology Satellite (ACTS) satellite and Internet emulators to measure the performance of the TCP protocol running under long-delay networks. The purpose of these studies was to identity a better TCP variant for use in long-delay networks such as the satellite environment and to investigate the effect of latency on aggregated network utilization. Similar to [1] and [2], we evaluated the protocol performance by analyzing the test results from our test-bed and realistic satellite link. Unlike [1] and [2], we have been concentrating on the performance of two protocol suites: File Transfer Protocol (FTP), the application-layer protocol that is running over TCP, and File Protocol (FP), an application-layer protocol running over Transport Protocol (TP) of the Space Communication Protocol Standards (SCPS) [4]

developed under the Consultative Committee for Space Data Systems (CCSDS) standards process. Another difference is that both [1] and [2] were done for satellites as part of the overall transmission link while we wish to terminate the data link in space, i.e., we consider satellites as regular Internet nodes. Our work has been directed at evaluating the performance of the protocols in a small satellite environment. We have developed a Space-to-Ground Link Simulator (SGLS) to provide the simulation capabilities to test both protocol suites [5], [6], [7], [8]. The satellite environment being simulated has one terminal at a ground station and the other terminal at the space vehicle. The link can either be direct broadcast or through a non-processing relay satellite such as the Tracking and Data Relay Satellite (TDRS).

The space environment poses a number of challenges to providing reliable, end-to-end data communication with a user-specified level of service. Losses due to transmission errors, large propagation delays, constrained bandwidth, asymmetric link rates, and intermittent connectivity all conspire to severely limit the performance of TCP. The goal of the CCSDS SCPS suite of protocols is to overcome these space channel problems and provide reliable data transport. The SCPS suite of protocols contains four elements: SCPS File Protocol (SCPS-FP), SCPS transport protocol (SCPS-TP), SCPS network protocol (SCPS-NP) and security protocol (SCPS-SP). SCPS-TP is actually TCP with a set of extensions and modifications aimed at improving TCP performance in space environment [9]. A large number of tests [5], [6], [7], [8] have been done to evaluate the performance difference between SCPS

and TCP/IP. All of the previous work was done over SGLS test-bed. These experiments might not provide the accurate performance of protocols in realistic satellite links due to the limitations of simulation approach and environment. In particular, the experiment results may not be applicable to networks with significantly different Bandwidth Delay Product (BDP).

This dissertation basically concentrates on studying the impacts of environmental changes on TCP performance and the performance differences between TCP SACK [10] and SCPS-TP over both the SGLS test-bed and satellite link. The goal of this study is to answer several basic questions, namely

(1) Is there an overall advantage of the SCPS protocol (SCPS-VJ or SCPS-Vegas) for file transport over TCP/IP in our simulated low BDP satellite channel? If there is an advantage, quantify it.

(2) Which congestion control option (VJ based or Vegas) can be invoked to improve protocol performance based on the performance comparison between SCPS-VJ and SCPS-Vegas?

(3) How do link delays and BER affect protocol performance?

(4) Does the SGLS simulator provide a reasonable (to within a scaling factor and offset) approximation to the true satellite channel, i.e., is there a linear translation between the two?

The above problems are specified in the form of hypotheses when we discuss the experimental work in section 3.2.

The rest of this dissertation is organized as follows. In Chapter 2, we describe the congestion control and loss recovery algorithms in TCP Tahoe [11], TCP Reno and TCP SACK, and TCP Vegas [12]. Chapter 2 also elaborates on the communication problems that TCP encountered in space environment and present TCP extensions to address these problems. Chapter 3 describes NMSU SGLS test-bed, the protocol software entities and the experimental procedures we use to conduct our experiments. Chapter 4 analyzes the behavior of both protocols over the SGLS test-bed by studying the averaged file transfer time of each file size for different configurations. Chapter 5 studies the behavior of both protocols over the realistic satellite link. Chapter 6 summarizes the dissertation and provides the conclusions and the future work for our study.

2 TCP PROTOCOL AND ITS EXTENSIONS FOR SPACE

TCP has several variants. Academia and industry generally make new ones by optimizing old ones based on experienced problems. This chapter briefly describes several of the most widely used variants of TCP. Based on the description, its extensions for space communications are provided.

2.1 TCP Protocols

A significant amount of today's Internet traffic, including World Wide Web (WWW) (HTTP), file transfer (FTP), email (SMTP), and remote access (Telnet), is carried over the modern TCP transport protocol. Actually, early TCP implementations followed by a go-back-*n* model using cumulative positive acknowledgments and requiring a retransmission timer expiration to re-send lost data. These TCP versions did little to minimize network congestion since the traffic congestion was not a serious problem at that time. Along with the explosive growth of the Internet, a variety of congestion avoidance schemes have been proposed to control network congestion while maintaining good user throughput. In this dissertation, TCP Tahoe refers to TCP with Slow Start, Congestion Avoidance, and Fast Retransmit algorithms [11] first implemented in 4.3 BSD Tahoe TCP in 1988. TCP Reno refers to TCP Tahoe with Fast Recovery implemented, first implemented in 4.3 BSD Reno TCP in 1990. TCP New Reno [13] is a modified version of TCP without SACK that avoids some of the TCP Reno performance problems when multiple packets are dropped from a window of data. TCP SACK refers to TCP Reno with SACK option added. TCP Vegas refers to the congestion control algorithm originally developed at

the University of Arizona in the x-kernel protocol framework by Lawrence Brakmo and Larry Peterson. The variants developed after TCP Tahoe can be categorized into three approaches, to avoid the bandwidth waste caused by inefficient loss recovery, to avoid the retransmission of successfully transmitted packets, and to avoid the periodic packet loss caused by self-generated congestion [14]. TCP Reno and New Reno belong to the first, SACK and Forward Acknowledgment (FACK) [15] belongs to the second, and TCP Vegas belongs to the third approach. Table 2.1 provides a summary of the major TCP variants.

Table 2.1: Summary of TCP variants

Variant	Major Algorithms	Document	Performance Problem
TCP Tahoe	Slow Start Congestion Avoidance Fast Retransmit	RFC 2001 [10]	Bandwidth waste by inefficient loss recovery—Channel tends to empty after Fast Retransmit and needs Slow Start to re-fill it after a single packet loss.
TCP Reno	TCP Tahoe + Fast Recovery	RFC 2001 [10]	Retransmission of successfully transmitted packets when multiple packets are lost from one window of data
TCP SACK	TCP Reno + SACK	RFC 2018 [3]	Period packet loss caused by self-generated network congestion (under study)
TCP Vegas	New Retransmission New Congestion Avoidance Modified Slow Start	[12]	Unfairness of bandwidth sharing (under study)

In general, we consider all TCP Tahoe, TCP Reno and TCP SACK to be Van Jacobson (VJ) congestion-control-based TCPs since they were developed based on three algorithms, Slow Start, Congestion Avoidance and Fast Retransmit, originally proposed by Van Jacobson in [11]. The widely used Tahoe, Reno and SACK, and Vegas are described in the following sections. Then we will examine how the space channel should interact with the protocols.

2.1.1 TCP Tahoe

TCP Tahoe added three new algorithms and refinements to earlier implementations. These three new algorithms are Slow Start, Congestion Avoidance, and Fast Retransmit. The refinements include a modification to the Round-Trip Time (RTT) estimate used to set retransmission timeout values.

Slow Start is the way to initiate data flow across a connection. In principle, Slow Start operates by observing that the rate at which new packets should be injected into the network is the rate at which the acknowledgments are returned by the other end [16]. Slow Start maintains two windows for each connection: the advertised window, called *awnd*, and the congestion window, called *cwnd*. The advertised window is flow control imposed by the receiver while the congestion window is flow control imposed by the transmitter. The former is related to the amount of available buffer space at the receiver for this connection; the latter is based on the sender's assessment of perceived network congestion [10]. The sender starts by transmitting one segment and waits for its ACK. When that ACK is received, the congestion window is incremented from one to two, and two segments can be sent. When each of

those two segments is acknowledged, the congestion window is increased to four. This provides an exponential growth in the window size if the ACKs are not delayed by the receiver and provides an approximately exponential growth typically because the receiver sends one ACK for every two segments that it receives. At some point the capacity of the link will be reached, and an intermediate router will start discarding packets. This tells the sender that its congestion window has gotten too large and the sender needs to slow down its transmission. At this point, the Congestion Avoidance algorithm is used to deal with lost packets.

Congestion Avoidance adds another variable, slow start threshold window size, called *ssthresh*, which is always set to half of the window size at the last packet loss except that its initialized value is 65,535 bytes. We see *ssthresh* changes depending on the window size at which a packet loss occurs. In the Slow Start phase, *cwnd* starts growing from one and grows rapidly for every successfully acknowledged packet until it reached *ssthresh*. In other words, it increases the window size to a value no larger than half of the size at which the packets loss occurred last time. Then TCP moves to the Congestion Avoidance phase and probes for extra bandwidth by increasing *cwnd* by 1/*cwnd* each time an ACK is received. From the above, we see Congestion Avoidance increases *cwnd* by at most one segment each RTT regardless how many ACKs are received in this RTT while Slow Start increases *cwnd* by the number of ACKs received in a RTT. Thus, Congestion Avoidance is an additive increase phase, compared to Slow Start's exponential increase. Although Slow Start

and Congestion Avoidance are independent algorithms with different objectives they have always been implemented together in current implementations.

Congestion Avoidance increments the current window size until another loss is detected. Congestion Avoidance assumes that packet loss caused by link error damage is very small, therefore the loss of a packet signals congestion somewhere in the network between the source and the destination. There are two indications of packet loss: a retransmission timeout occurring and the receipt of duplicate ACKs. If three or more duplicate ACKs are received for the same TCP segment, the transmitter considers it as a strong indication that a segment following the acknowledged one has not been received properly since the receiver has kept acknowledging a particular one instead of the next one the transmitter sent. In this case, the transmitter should retransmit only one segment starting with the sequence number just acknowledged without waiting for a retransmission timer to expire. This is how Fast Retransmit works. This loss specifies the point at which another cycle begins, i.e., TCP needs to go back to Slow Start and repeats the above cycle. Clearly, Fast Retransmit leads to higher connection throughput and better channel utilization than that the transmitter waiting for the retransmission timer expiring. The details of Slow Start, Congestion Avoidance, and Fast Retransmit are described in [10] and [11].

Basically, if we call the current window size W and allow *ssthresh* to equal W_{th}, then the behavior of TCP Tahoe can be simplified by updating parameters W and W_{th} as follows:

Set W=1 segment and W_{th}=65,536 bytes.

- For every acknowledgment that arrived at the transmitter:

  If $W<W_{th}$, running Slow Start algorithm with $W=W+1$.

  If $W\geq W_{th}$, running Congestion Avoidance with $W=W+1/\lceil W \rceil$.

- For a packet loss is detected:

  $W_{th}= W/2$;

  $W=1$.

2.1.2 TCP Reno

TCP Reno retains the enhancements incorporated into Tahoe but modifies the Fast Retransmit algorithm to include another algorithm, Fast Recovery [11]. Continuing the data transfer situation in 2.1.1, if the Congestion Avoidance algorithm is followed, instead of Slow Start, after the Fast Retransmit operation when a loss occurs for TCP Tahoe, it is known as Fast Recovery. The reason for not using Slow Start in this case is that the receipt of a number of duplicate acknowledgments tells the transmitter not only a packet has been lost but also that the data can still flow on the channel. Thus, the transmitter should not reduce the flow abruptly by stepping into the Slow Start phase. This algorithm operates under the assumption that each duplicate acknowledgment implies a single packet has left the transmission channel. Clearly, Fast Recovery prevents the communications channel from tending to empty after Fast Retransmit, thereby avoiding the need of Slow Start to re-fill it after a single packet loss. With Fast Recovery, the transmitter is able to make an intelligent estimate of the amount of outstanding data.

The TCP Reno Fast Recovery algorithm is optimized for the case of a single packet loss out of the whole window data. This means the transmitter need retransmit only one packet per RTT. This significantly improves the performance of TCP Tahoe for a single loss case, but can suffer when multiple packets are lost out of a window of data. But since single packet loss is considered to be the predominant loss in space, the Fast Recovery algorithm is expected to work well over satellite link.

A simplified description for understanding the main idea of TCP Reno is shown below. Similar to TCP Tahoe, it concentrates on updating two parameters, current window size W and Slow Start threshold W_{th}, in different ways corresponding to in different phases.

Set W=1 segment and W_{th}=65,536 bytes.

- For every nonrepeated acknowledgment that arrived at the transmitter, TCP Reno operates in the same as TCP Tahoe

  If $W<W_{th}$, $W=W+1$ during Slow Start Phase

  If $W\geq W_{th}$, $W=W+1/\lceil W \rceil$ during Congestion Avoidance Phase

- When the duplicate acknowledgments exceed a threshold

  1. Retransmit missing segment;
  2. Set $W_{th}= W/2$ and let $W=W_{th}$;
  3. Resume Congestion Avoidance using the new window size once retransmission is acknowledged.

- Upon retransmission timer expiration, enter the Slow Start algorithm and operate as follows

Set W_{th}= W/2 and W=1.

A *cycle* is defined to represent the TCP evolution starting from the end of one Congestion Avoidance phase to the end of the next. For a typical TCP Reno session, each cycle begins when a loss is detected using duplicate acknowledgments. We also see that Slow Start is not involved after an initial Slow Start transient since the window size is already halved upon loss detection. If W_{max} is the window size at which a loss occurs, each cycle begins at the window size of W_{max}/2. The algorithm resumes probing for extra bandwidth in Congestion Avoidance mode until the window size reaches W_{max} again, at which point a loss occurs and another cycle begins with window size of W_{max}/2. W_{max} can also be considered as a generic notation of window size at which Congestion Avoidance ends since the algorithm exits the Congestion Avoidance phase when a loss occurs. W_{max} varies in time and its size could be different in each cycle if the losses occurred are random, or could be identical for losses occurred periodically.

TCP New Reno is actually TCP Reno with a slight adjustment at the transmitter that eliminate Reno's wait for a retransmit timer for multiple losses case [17].

2.1.3 TCP SACK

Noting the problem suffered by TCP Reno when multiple losses occur, TCP SACK is designed to overcome this problem by combining Reno with a selective repeat retransmission policy [3]. In principle, the receiving TCP sends back SACK packets informing the sender about all data packets that have been received

successfully, so the sender need retransmit only the segments that have actually been lost.

TCP SACK enters the Fast Recovery phase as Reno does for duplicate acknowledgments that arrived at the transmitter. The difference is that, during Fast Recovery, SACK maintains another variable, *pipe*, that is used to estimate the number of packets outstanding in the transmission channel. The sender only sends new or retransmitted data packets when *pipe* is less than the congestion window. The variable *pipe* is incremented by one when the sender either sends a new packet or retransmits an old packet. It is decremented by one when the sender receives a duplicate ACK packet with a SACK option reporting that new data has been received by the receiver.

The use of *pipe* decouples the decision of *when* to send a packet from the decision of *which* packet to send. A data structure called a *scoreboard* is maintained at the sender and it remembers acknowledgments from previous SACK options. When the sender is allowed to send a packet, it retransmits the next packet from the list of packets inferred to be missing at the receiver. If there are no such packets and the receiver's *awnd* is sufficient large, the sender transmits a new packet. The sender leaves Fast Recovery phase when a recovery ACK is received acknowledging all data that was outstanding when Fast Recovery was entered. The details of TCP SACK can be referred to [3] and [18]. There is some empirical evidence in favor of the superior performance of selective acknowledgment. Simple experiments [3] showed that

disabling the selective acknowledgment greatly increases the number of retransmitted segments over a lossy, high-delay Internet link.

2.1.4 TCP Vegas

TCP Vegas is a congestion avoidance scheme designed to prevent the periodic packet loss that occurs in other algorithms. It successfully reduces queuing and packet loss, and thus reduces latency and increases overall throughput than Reno by matching the sending rate to the rate at which packets are successfully being drained by the network.

Reno assumes that the loss of segments was due to congestion regardless of what the causes really are, congestion, corruption, or link outages, and cuts the data rate in half, and then gradually increases it until another loss situation occurs and repeats this process until all data have been transmitted. Clearly, Reno has no mechanisms to detect the incipient phases of congestion before losses occur and hence cannot prevent such losses. On the contrary, TCP Vegas tries to sense incipient congestion by observing the variations of RTT or the actual variations of throughput. Since TCP Vegas infers the congestion window adjustment from such throughput measurement, it may be able to slow down the data rate before the congestion induces loss. Therefore, the VJ based congestion control method is "reactive," as it waits to know the available bandwidth at the cost of packet loss while TCP Vegas congestion detection mechanism is "proactive" [19], [20]. The TCP Vegas congestion control was chosen to be the default congestion control algorithm of SCPS-TP since it facilitates differentiating between losses due to congestion and those due to

corruption. But SCPS-TP's congestion control algorithm can be set either to VJ congestion control or to TCP Vegas by an application based on different assumptions of the source of packet loss. In our experiments, SCPS protocol is simply called SCPS-VJ or SCPS-Vegas depending on SCPS-TP's congestion control algorithm is set to VJ congestion control or to TCP Vegas. Although they are considered as two protocols in this study, SCPS-VJ and SCPS-Vegas are actually the same protocol running two different congestion control modes.

The following paragraphs explain in detail how TCP Vegas works differently by adopting a new congestion detection and control mechanism based on [12]:

New Retransmission Mechanism: For Reno congestion control, the RTT estimate is not very accurate since it is estimated using a coarse-grained timer. This coarse granularity influences both the accuracy of the estimate itself and the frequency at which TCP checks to see if a segment should be timed out. As mentioned in Section 2.1.1, there are two indications of packets loss: a timeout occurring and the receipt of duplicate ACKs [16]. Reno needs to retransmit the packets lost in the above situations. In comparison, the TCP Vegas congestion control algorithm introduces three major modifications to Reno retransmission policy. First, TCP Vegas measures the RTT for every segment sent using a fine-grained system clock and a timeout period for each segment that is computed using this more accurate RTT estimate. For a duplicate acknowledgment situation, it checks to see whether the timeout period has expired. If so, it retransmits the segment without having to wait for three or more duplicate ACKs which is required for Reno to

retransmit. Second, when the first or second non-duplicate ACK after a retransmission is received, the TCP Vegas again checks for the expiration of the timer. If it is expired, then retransmits the segment. This will catch any other segments that may have been lost prior to the retransmission without waiting for a duplicate ACK. Third, in case of multiple segment loss and more than one fast retransmission, the congestion window needs to be reduced only for the first fast retransmission. Any losses that occurred before the last window decrease were not inferred as the network congestion for the current congestion window size, and therefore, there is no further window size decrease. This modification is required since the TCP Vegas needs to detect losses much earlier than Reno congestion control algorithm.

Congestion Avoidance Mechanism: As mentioned above, the TCP Reno congestion detection and control algorithm use the loss of segments as a signal that congestion is occurring in the network. It has no mechanism to detect the precursory phases of congestion and to prevent losses of segments before it really happens. Thus, it is "reactive" or "passive." In comparison, the TCP Vegas algorithm tries to detect incipient congestion by comparing the estimated RTT to the expected RTT. The congestion window is increased only if these two values are close. This implies that the algorithm will increase the window size only if the capacity of the network is large enough to achieve the expected high throughput (or the expected short RTT). If the estimated RTT is considerably longer than the expected one, then the congestion

window should be reduced. This can be considered as a sign of incipient congestion occurring.

Modified Slow-Start Mechanism: TCP Reno has a high cost in terms of too many segment losses if there is no an appropriate limit for the size of the congestion window or if it has a very fast sender. Since the size of the congestion window is doubled for every RTT before losses occur, which is equivalent to doubling the attempted throughput every RTT, the loss of packets is expected to be on the order of half the current congestion window at some point when the available bandwidth is finally overrun, and even worse if a traffic burst from another connection is encountered. Comparing with Reno algorithm, TCP Vegas algorithm incorporates a similar congestion detection mechanism into the slow-start phase to determine when to change to the congestion avoidance phase. In order to detect and avoid congestion during the slow-start phase, the congestion window is allowed to grow exponentially only for every other RTT. The congestion window is fixed in between to have a valid comparison of the expected and the actual rate. When the actual rate falls below the expected rate by the equivalent of one route buffer, Vegas changes from exponential-increasing Slow Start phase to linear-increasing Congestion Avoidance phase.

The reason to measure the actual rate with a fixed congestion window is that we want actual rate to represent the bandwidth allowed by the connection. Thus, we can only send as much as data as is acknowledged in the ACK (during Slow Start, TCP Reno sends an extra segment for each ACK received). This modified Slow Start

mechanism is very successful at preventing the losses incurred during the initial Slow Start period [12].

In [12], Brakmo claimed that TCP Vegas is able to achieve between 40 and 70% better throughput, with one-fifth to one-half the losses, as compared to the implementation of TCP Reno. By decomposing the TCP Vegas algorithm into its various mechanisms and assessing the effect of each of these mechanisms on performance, [19] indicates that the above performance gains are achieved primarily by the techniques used in Vegas for slow-start and congestion recovery. Its congestion avoidance mechanism is shown to have only a minor influence on throughput. Other work [14] shows that Vegas does not concern the fairness among source-destination pairs with different RTTs. The fairness sharing link resource is an important network performance factor when multi-users access the network. It nevertheless should be concerned, especially for a Wide Area Network (WAN).

Figure 2.1 shows the relationships between widely used TCP Tahoe, Reno and SACK with respect to the transitions of Slow Start and Congestion Avoidance phases.

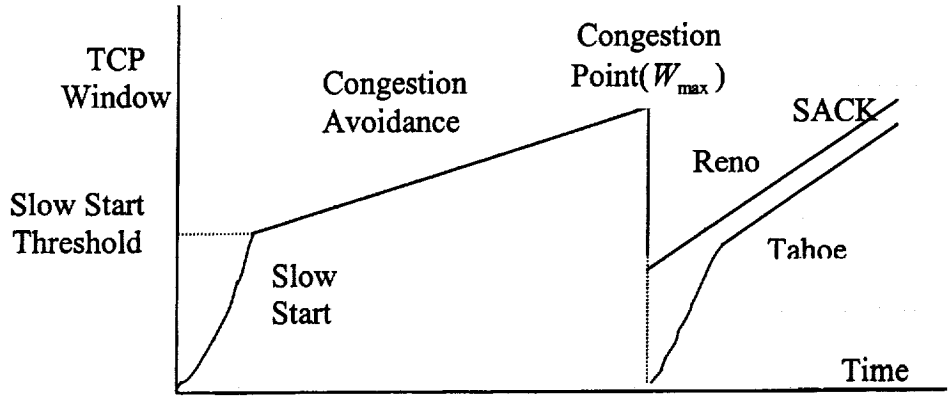

Figure 2.1: Widely used TCP variants

2.2 TCP Problems in Space

Satellite channels are dominated by two fundamental characteristics, noise and bandwidth [24], that impede reliable data communications. The following paragraphs describe features related to the above two major characteristics. With the various constraints of in-lab simulation, some features described below could not be simulated on our SGLS test-bed. In particular, we simulate a satellite link with a relatively low Bandwidth-Delay Product (BDP) or capacity only.

2.2.1 High Transmission Errors

Transmission error rate caused by noise in space communication is much higher than that on terrestrial networks. The Bit-Error-Rate (BER) for most space channel is around 1E-6 to 1E-5 while the BER for widely used optical channel on the ground is only about 1E-12. Many more packet losses occur due to higher transmission error in space. TCP assumes that all packet losses were due to congestion regardless of what the causes really are, congestion, corruption, or link outages. This makes TCP activate its congestion control algorithm, reduces its congestion window, and finally, decreases its throughput. We see TCP congestion control schemes work well in dealing with congestion-induced loss, but results in reduced throughput on noncongestioned, noisy links without providing any benefits [9].

2.2.2 Asymmetric Channels

Communications channels between spacecraft and the ground are frequently asymmetric in terms of both channel capacity and error characteristics [9]. In general,

the return link bandwidth (from the spacecraft to the ground) is substantially larger than the forward link bandwidth. The asymmetry is related to the fact that the forward link is generally used for commanding the spacecraft (not bulk data transfer) while the return link is used to flow the data generated at the satellite to the ground. The high asymmetry of satellite link bandwidth is not a property shared by terrestrial networks. This can limit TCP throughput even when high bandwidth link flows data and lower bandwidth link carries the acknowledgments. Following the principle of TCP congestion control, the new data transmission rate is proportional to the acknowledgment rate returned by the receiver, so TCP performance is limited by low bandwidth of acknowledgment channel.

2.2.3 Overhead of TCP Protocol Header

Wireless channels tend to provide less available bandwidth than terrestrial networks. In space environment, this problem is coupled with the constraint that transmission power is limited and bit-efficiency is important in terms of the cost of transmitting as well as in terms of link capacity.

The substantial bit overhead of TCP protocol header is not beneficial with respect to the scarcity of available bandwidth in space. This is especially inefficient when transmitting small data segments.

2.2.4 Large Bandwidth-Delay Product

Bandwidth-Delay Product (BDP) or the capacity of the pipe between the transmitter and the receiver can be calculated as

$$\text{Capacity (bits)} = \text{Bandwidth (bits/second)} \times \text{RTT (second)}.$$

BDP represents the amount of data in flight or the amount of data that would fill the pipe. In other words, it defines the total unacknowledged data that can be injected into the network to keep the pipeline full. It is actually the buffer space required at the sender and receiver to achieve maximum throughput on the TCP connection over the communication path. Long RTT makes BDP being very large. This requires TCP to keep a large number of packets outstanding. This may not be a problem for a low BDP satellite system like our simulator but is a problem on large satellite systems since current TCP in terrestrial networks was not developed to work in a large BDP environment.

2.2.5 Intermittent Connectivity and Variable RTT

Intermittent connectivity and the variable RTT of the space channel cause an unstable flow of acknowledgments and inaccuracy in determining the RTT. The former makes TCP invoke its congestion control and retransmit packets frequently and the latter causes unnecessary invocation of the Slow Start algorithm. Both effects reduce the throughput.

2.3 TCP Extensions for Space Communications

This section briefly describes the proposed solutions that SCPS-TP attempts to overcome the above problems in space with the improvements to TCP. Considering the constraints of our simulation mentioned above, the advantages of some of the following solutions could not be displayed in our experiments.

2.3.1 Overcoming Losses from Different Sources

There exist at least three sources of loss in space environment, network congestion, corruption, and link outage. SCPS-TP responds to each of them differently. SCPS-TP has two mechanisms for determining the source of packet loss. Unlike TCP (default assumption is that all loss is caused by congestion), SCPS-TP uses a parameter to set the default assumption of the source of packet loss on a per route basis in the absence of any explicit information. Based on these different assumptions, SCPS-TP's congestion control algorithm can be set either to VJ congestion control or to TCP Vegas by an application.

2.3.1.1 Congestion-Induced Loss

TCP Vegas is adopted to be the default congestion control mechanism in SCPS-TP to minimize loss and facilitate the use of large window. Vegas does not depend on the receiver window as an upper bound on the size of the congestion window. It bounds itself and avoids network congestion without overdriving the link to find the saturation point. SCPS-TP modifies Slow Start algorithm by providing an additional trigger for transitioning from the congestion window's exponential growth phase into the linear growth Congestion Avoidance phase.

2.3.1.2 Corruption-Induced Loss

In the case that the packet loss is caused by transmission errors instead of congestion, SCPS-TP uses an open-loop, token bucket rate control mechanism [21] to keep from overflowing the link capacity instead of invoking congestion control in response to packet loss. Token bucket rate control mechanism meters out the

transmissions at a specified rate. The allowed transmission rate for each link is a managed parameter that is stored in the globally accessible routing structure at each endpoint. On a host, the available capacity for a particular link is shared among all SCPS-TP connections using that link.

2.3.1.3 Link Outage Loss

If a host running SCPS-TP receives a link outage signal from another SCPS-TP host, the correct response is to enter persist mode, sending periodic probe packets. It does not repeatedly time-out, retransmit, and back-off the retransmission timer.

2.3.2 Coping with Asymmetric Channels

The SCPS-TP receiver delays acknowledgments for a configurable period of time that is related to its estimate of the RTT instead of acknowledging at least every other segment or acknowledging immediately. SCPS-TP also uses header compression to reduce significantly the overhead on the acknowledgment channel and get higher acknowledgment rates. This header compression is different from the TCP/IP header compression scheme. See Section 2.3.3.1.

2.3.3 Relieving Bandwidth Constraints

Header Compression and Selective Negative Acknowledgment (SNACK) are two mechanisms to improve SCPS-TP performance in bandwidth-constrained environment.

2.3.3.1 SCPS-TP Header Compression

SCPS-TP does not use RFC 1144 TCP/IP header compression [22] because this header compression was designed for use on low-speed serial link and is

performed on a hop-by-hop basis at the link layer. SCPS-TP uses a loss-tolerant TCP header compression scheme that operates end-to-end, at the transport layer, and can tolerate loss and changing connectivity. By operating end-to-end, SCPS-TP header compression avoids the problems caused by changing connectivity, since transmitter and receiver, where the compressor and decompressor reside, never change. But if satellite telemetry is basically single-hop, then this should make no real performance difference.

2.3.3.2 SCPS-TP SNACK

SCPS-TP SNACK option draws from both TCP SACK option and TCP Negative Acknowledgment [23]. Like NAK, SNACK is a negative acknowledgment, but it is capable of specifying multiple holes in the sequence space buffered by the receiver in a bit-efficient manner. By providing more information about lost segments more quickly, SNACK option can hasten recovery and prevent the sender form becoming window –limited, thus allowing the pipe to drain while waiting to learn about lost packets. The ability to keep transmitting in the presence of packet loss is especially important when loss is caused by corruption rather than congestion. In this case, SNACK is of particular benefit in keeping the pipe full and allowing transmission to continue at full throttle while recovering from loss. In a low BER case where the channel is mostly single packet loss, SNACK may not be very useful. The details of SNACK can be found from [9].

2.3.3.3 Other Techniques for Coping with Errors

Besides the above mechanisms, SCPS-TP also employs two other techniques that TCP uses: Timestamps option and TCP Window Scaling option [24].

Many current TCP implementations with the time-stamp option disabled base their RTT measurements upon a sample of only one packet per window. This method of timing one segment per window yields an adequate approximation to RTT for connections with low bandwidth-delay products, but results in an unacceptably poor RTT estimate when the bandwidth-delay product of the network grows [24]. The TCP Timestamp option lets the sender place a timestamp value in every segment. This helps TCP make accurate RTT estimates in the face of loss, when it can be difficult to time the round-trip of particular segments that may be lost and subsequently retransmitted in a "long, fat pipe" network (LFN) [25].

The 16 bit receiver window size of TCP header limits the largest window that can be used to be 65,536 bytes. TCP performance problems arise with 65-Kbyte receiver windows in an "LFN". The Window Scaling option expands the TCP window size from 16 to 32 bits to permit TCP to have more than 64 Kbytes of data outstanding at one time. This was done simply by imposing an implicit scale factor on the advertised window instead of changing TCP header size. Such a large window will allows the sender to continuously send new data while retransmitting lost segments, even as the left edge of the window does not advance for periods of time.

In this dissertation, we compare the performance of SCPS-TP to that of regular TCP. The comparison is done by running ftp over TCP/IP stack and SCPS-FP

over SCPS-TP/IP stack on both the SGLS test-bed at the Center for Space Telemetry and Telecommunications of NMSU and Telsat II satellite link operated by Loral Skynet. SCPS-TP we tested comes with SCPS version SCPS_RI 1.1.48 provided by MITRE. The implementation of TCP that we use is the default TCP incorporated into Red-Hat Linux 6.1(kernel version 2.2.12-20). Red-Hat Linux 6.1 TCP supports all the following TCP algorithms implemented in TCP Tahoe, TCP Reno and TCP SACK as mentioned in Section 2.1: Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery and SACK.

Based on the introduction of congestion control algorithms in TCP variants and its extensions in space environment in this chapter, Chapter 3 describes the SGLS test-bed, experimental assumptions, experimental procedures we use to conduct our tests to compare the performance of TCP and SCPS-TP. The details of the protocol configurations and test hypotheses for our experiments are also included.

3 SGLS TEST-BED AND EXPERIMENT METHODOLOGIES

This chapter describes NMSU SGLS test-bed facility, the procedures and the protocol entities we have used to conduct the experiments.

3.1 Tests over Simulated Test-bed

The SGLS channel simulator is used to perform the error generation and link delay used to test the protocol suite performance. In the following subsections, we introduce the simulator and discuss the experiment methodologies we have used to conduct the tests.

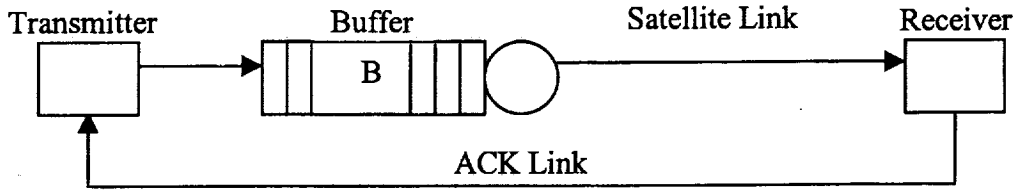

Figure 3.1: A typical satellite link model

3.1.1 Channel Simulator

A typical satellite link model is given in Figure 3.1. Basically it consists of the transmitter, receiver, link buffer and satellite link. The Space-to-Ground Link Simulator (SGLS) has been developed at NMSU to model space channel characteristics experienced in transmitting data. The simulator is described fully in [5], [6], and [7]. Basically, the SGLS configuration allows the user to configure the simulated channel to

- Allow for simultaneous bi-directional data flow (forward and return channels),
- Allow user-selectable error rates and statistical descriptions of the channel,

- Allow different data rates on the forward and return links as would be found in satellite links, e.g. 2400 baud forward, 57,600 baud return, and
- Provide for a simulated delay up to 5 seconds on each link.

The SGLS utilizes the LabVIEW programming language to control data throughput through the simulator, mix the baseband data stream with the user-selected error vector, and provide for the user-selectable link delay value. The hardware configuration is illustrated in Figure 3.2. The LabVIEW software is run as an application on each of the SGLS computers. Typically, the LabVIEW modules are the only applications software running on the computers. This configuration was developed to model point-to-point satellite links in its current configuration. The bandwidth-delay product for the system under a 57,600 bps symmetric link with no imposed channel delay is 671 bytes. As a comparison, a T-1 line crossing the United States has an estimated bandwidth delay product of 11,580 bytes. Therefore, this simulator corresponds to a relatively low BDP system.

The three PCs in the SGLS are Dell 600-MHz computers with 128 Mbytes of memory running Windows 98 second edition. The first Linux-based PC is a Dell 266 MHz computer. This is our logical ground station computer. The second Linux-based PC is a Gateway 166 MHz. This is our logical satellite computer. Both Linux computers are running Red Hat Linux version 6.1. The SGLS is connected to the Linux computers using serial cables connected to the COM serial ports on each computer. The data connections are configured without hardware or software handshaking to allow for a simulation that would be similar to interfacing with a

satellite radio system. In all cases, the links between the SGLS and the Linux computer were set to 57,600 bps (R2 in Figure 3.2). The simulations run with symmetric links had the forward link (R1 in Figure 3.2) also set to 57,600 bps. The simulations run with asymmetric links had the forward link set to 2400 bps. Other combinations are possible and the reader should refer to [8] for representative results.

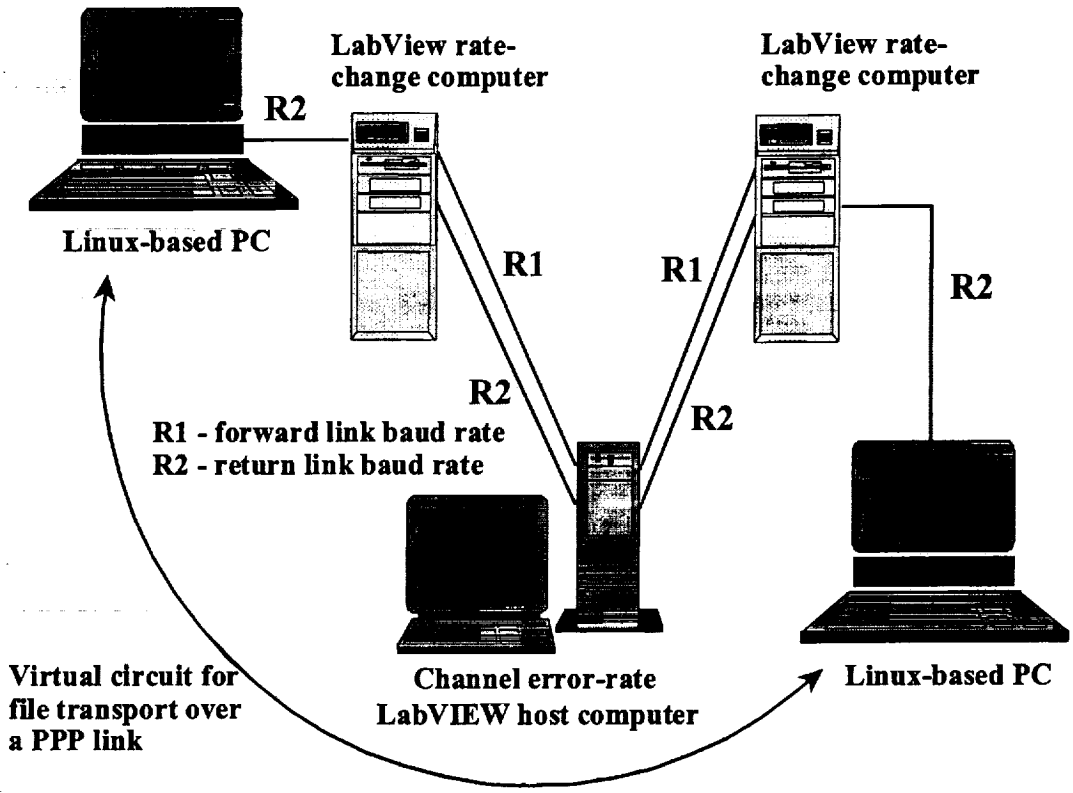

Figure 3.2: SGLS hardware configuration

3.1.2 Experiment Tools Used in NMSU Test-bed

The experiments run at NMSU benefit from several software tools for control and analysis. The following subsections describe these tools.

3.1.2.1 Expect Scripts to Automate Tests

Two Expect scripts were modified based on models provided by MITRE. They were developed to automate SCPS-FP and FTP tests in NMSU test-bed. They basically achieve the following objectives:

- Automate file transfers and gather reported transmission times for multiple runs at different file sizes for both protocols in one experiment configuration;
- Capture traffic performance over Point-to-Point (PPP) interface for each connection for both protocols using existing tool `tcpdump` that is supplied with Red Hat Linux;
- Conduct tests above with various error rates under human intervention to set the BER in the SGLS;
- Achieve all the above three objectives over different interfaces (e.g., Ethernet, ATM) after trivial modifications.

3.1.2.2 Tcpdump, Tcptrace and Xplot

Tcpdump, tcptrace and xplot are the major tools that have been used in NMSU testbed to observe and analyze TCP/IP and SCPS performance. Each is described below.

Tcpdump is a packet capture program. Basically it prints out the headers of packets over a network interface. In our test-bed, we have used it to dump the traffic over PPP interface to obtain binary data files for both protocols. Those data files would then become the sources from which the performance knowledge can be obtained by cooperating with two tools. Tcpdump is supplied with the Red Hat

distribution software and is also publically available via anonymous ftp from *<ftp://ftp.ee.lbl.gov/tcpdump.tar.Z>*.

Tcptrace is a TCP dump file analysis tool. It, in general, tells us the detailed information about TCP connection by sifting through dump files. It reads output dump files in the formats of several popular packets capturing programs: tcpdump, snoop, etherpeek and others, and produce different types of performance graphs. See Section 3.1.7 for more information.

Xplot is a plot tool. In our system, it is used to plot various performance graphs made using tcptrace. Xplot and tcptrace tools may be obtained from *<http://www.tcptrace.org/>*.

3.1.3 Protocol Layers and Configurations

The following subsections discuss the protocols layers and configurations for the tests done in our test-bed and satellite link.

3.1.3.1 Protocol Software Entities

The software used in these experiments is used "as is" from the suppliers without any attempt to modify it. The only changes are to select options as described in the experiments. It is felt that this would most closely resemble the situation used by most system developers who are more concerned with satellite development than attempting to fine tune software.

The operating system used on the source and destination data computers is Red Hat Linux version 6.1. The kernel build is 2.2.12-20.

The TCP/IP and data link layer PPP protocols are those that come with the Red Hat installation software. Both are installed in the kernel without modification. As mentioned in the end of Chapter 2, Red-Hat Linux 6.1 TCP supports all the following TCP algorithms implemented in TCP Tahoe, TCP Reno and TCP SACK: Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery and SACK.

The SCPS protocol suite is based upon the software provided by MITRE. All tests used to analyze the performance of SCPS-TP in this effort were conducted with version 1.1.48 of the SCPS RI software. Most of the previous work [8] was done with earlier version SCPS RI 1.1.34.

3.1.3.2 SCPS and TCP/IP Protocol Layers

As an application process, the SCPS Reference Implementation (RI) operates outside the Unix/Linux kernel. It uses the kernel's socket interface to bypass the transport and network protocols in the kernel and provide access to the network interfaces. To allow flexibility in the development and execution of SCPS-based applications, the SCPS Reference Implementation may operate over many different types of protocols and encapsulation mechanisms. This can be one by performing different configuration actions according to the users' needs before building the SCPS RI. Figure 3.3 illustrates the entire SCPS protocol stack and shows the various configuration options at the different layers in the SCPS protocol suite. The application layer FTP runs over TCP and the application layer SCPS-FP runs over SCPS-TP which is the extended version of TCP. Both TCP and SCPS-TP run over network layer protocol IP running over either PPP or Ethernet data link. For our

experiments over SGLS test-bed, the protocol suites we test are: FTP/TCP/IP/PPP for TCP/IP and SCPS-FP/SCPS-TP/IP/PPP for SCPS; and for the experiments over satellite link, the protocol suites are: FTP/TCP/IP/Ethernet for TCP/IP and SCPS-FP/SCPS-TP/IP/Ethernet for SCPS.

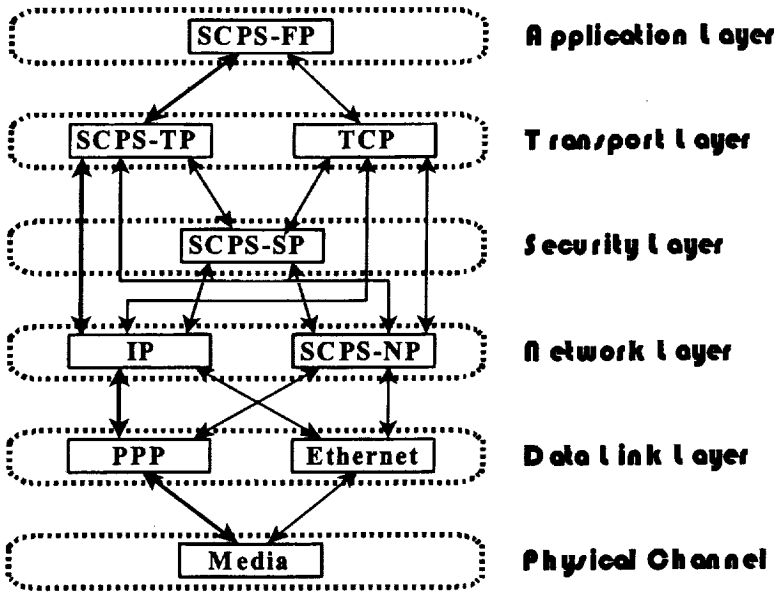

Figure 3.3: Protocol layers of SCPS and software entities

3.1.3.3 Protocol Configurations

This section briefly describes how congestion control options, header compression, timestamp option, window scaling option and acknowledgment options are configured with both protocols in our experiments.

Congestion Control

As mentioned, TCP/IP protocol suite is tested with the implementation of default TCP incorporated into Red-Hat Linux 6.1. This implementation is considered to use VJ based congestion control algorithms that were discussed in detail in Section

2.1. SCPS protocol suite is tested with both VJ and Vegas based congestion control modes that correspond to SCPS-VJ and SCPS-Vegas variants of SCPS implementation. The details of this issue are provided in [6], [7], [8] and will also be discussed more in Section 3.2.1.

Header Compression

During our experiments, we were careful to ensure that we compared SCPS and Linux's TCP implementation in as fair a manner as possible. One thing that we did not anticipate was that when the simulator was configured to have a very low speed acknowledgment channel, the presence or absence of TCP header compression greatly affected performance.

Our original tests (with TCP header compression enabled) showed great disparities in performance between ftp (using TCP) and SCPS-FP (using SCPS-TP). MITRE suggested that the difference was due to TCP header compression [22] in the Linux PPP driver. Specifically, IP datagrams entering into PPP need go through a compressor in the PPP driver. This compressor recognizes if the incoming packets are of the TCP protocol type by checking an 8-bit value of the protocol field in the IP header. Different Transport layer protocols, TCP, UDP, ICMP or IGMP all can send data to the IP layer. IP adds this field to the IP header to have an IP protocol number to recognize the protocol from which the data comes. IP protocol #6 indicates TCP, 1 is for ICMP, 2 is for IGMP and 17 is for UDP [16]. By default, the compressor in the PPP driver compresses TCP/IP headers (with IP protocol #6) from around 40 bytes down to 3-5 bytes, but it does not recognize (and hence does not compress) SCPS-TP

headers which are IP protocol #106 (and also about 40 bytes). To ensure that header compression was indeed the source of the performance difference, we ran our tests for ftp and SCPS-FP both with header compression turned ON and with it OFF. Running with both SCPS-FP and TCP with header compression turned OFF provided a fair comparison of the two protocols.

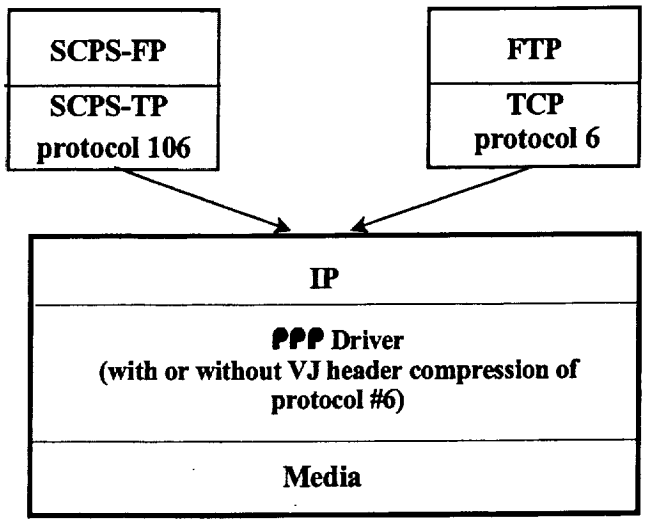

Figure 3.4: Flow of packets over PPP for both protocols

Figure 3.4 illustrates the flow of packets using both SCPS-FP and ftp. Note that at the level of the PPP driver, both SCPS-TP and TCP packets are encapsulated inside IP packets. The relevant difference is that TCP/IP packets contain an IP protocol ID of 6 while SCPS-TP/IP packets contain an IP protocol ID of 106. The VJ header compression machinery in the PPP driver operates ONLY on IP packets whose protocol ID is 6. Thus SCPS-TP/IP packets transmitted over the PPP link are always sent uncompressed. For TCP/IP packets, the PPP driver can be configured to either use or bypass VJ compression.

While we did not use it in our tests, it is worth noting that SCPS-TP can be configured to use its own header compression that is fundamentally different than the standard Van Jacobson (VJ) header compression. While VJ header compression operates on a link-by-link basis, compressing and re-expanding TCP headers each time they are transmitted/received, SCPS-TP header compression operates end-to-end. Also, VJ header compression uses a method known as differential or delta encoding, whereby changes to certain fields of the TCP header are communicated by sending the difference between the current value and the last value sent. This method depends on correct reception of the N^{th} TCP segment in order to correctly decompress the $(N+1)^{st}$ segment. If a single TCP segment is lost, all subsequent segments will fail to decompress until an uncompressed segment, typically a retransmission of the first segment lost is sent, and are lost. This means that a single packet lost on a link that is using header compression generally forces a retransmission timeout (RTO) in order to recover.

Reducing the forty-byte packet header size to only three bytes greatly reduces the interactive response time and increases the line efficiency. We should note that the VJ header compression was especially made to improve TCP/IP performance over low speed (300 bps to 19,200 bps) serial links. In large bandwidth-delay product networks with moderate to high error rates, we expected that VJ header compression actually decrease the performance. SCPS-TP header compression, by contrast, does not use differential encoding. This results in a slightly lower compression ratio (compressed SCPS-TP headers are typically slightly larger than VJ- compressed TCP

headers) but increased robustness, as a single lost packet **does not** cause the loss of subsequent packets. The single lost packet can then be recovered via standard means (SNACK, fast retransmit, etc.) and the sending TCP will hopefully **not** have to halve its transmission rate. To fairly compare the two header compression schemes, we would need to compare TCP/IP with VJ header compression and SCPS-TP with header compression running over the SCPS network protocol, SCPS-NP. To date we have not done this and it is left as future work.

In our previous work [8], only VJ header compression was disabled. Under MITRE's suggestions, three other types of PPP compressions were also disabled for the tests over the SGLS test-bed:

- **deflate**-This is the default bulk data encryption algorithm for serial links. It requests that the peer compress packets that it sends.
- **bsdcomp**-It is another bulk data encryption for serial links. It is used to request that the peer compress packets that it sends using different compress scheme.
- **predictor1**-Functions the same as the above two using different scheme.

The test results over the SGLS test-bed used for our analysis in this dissertation were obtained by disabling VJ header compression and the above three PPP compressions. For the test results and analysis of the impacts of VJ header compression on protocol performance, see [8]. The impacts of **deflate, bsdcomp**, and **predictor1** compressions have not been analyzed in detail.

Timestamps

Earlier TCP implementations with the time stamps option disabled base their Round Trip Time (RTT) measurements upon a sample of only one packet per window. This method of timing one segment per window yields an adequate approximation to the RTT for connections with low bandwidth-delay products, but results in an unacceptably poor RTT estimate when the bandwidth-delay product of the network grows [24]. The TCP timestamp option lets the sender place a timestamp value in every segment. The receiver reflects this value in the acknowledgment and allows the sender to calculate an RTT for "each received ACK," which is used to calculate a Retransmission TimeOut (RTO) value. In order to achieve high performance and reliable operation in a "long, fat pipe" network (LFN), the most current and accurate RTT and RTO estimates possible are necessary to adapt to changing traffic conditions. The "correct" value of the RTO may change during the course of a connection if the RTT changes significantly.

Both TCP and SCPS TP are symmetric protocols, which allow user data to be sent in either direction on a single connection. Therefore timestamp always be sent and echoed in both directions as per RFC 1323 Timestamps [24]. By default, the time stamp options are enabled for both Red Hat Linux 6.1 (with kernel 2.2.12-20) built-in TCP/IP and SCPS-TP.

In details, enabling or disabling the time stamp option will have the following impacts:

- Enabling the time stamp options adds an additional 12 bytes of overhead to the TCP/IP and the TP/IP segment header of 40 bytes. In other words, for a frame size of 1500 bytes over the PPP link in our system, the default time stamp being enabled reduces the available data size from (1500-40)=1460 bytes to (1500-52)=1448 bytes.
- Since the TCP and SCPS-TP time stamps work symmetrically, enabling them also adds 12 bytes to the TCP and TP header of every ACK. Since a "plain" ACK (on a connection that is not using the time stamps option) is 40 bytes long, the 12 bytes of time stamp make a considerable difference on low-rate acknowledgment channels. For example, on the 2400-bps acknowledgment channel in our asymmetric tests, we can send a maximum of (2400/8)/40 = 7.5 ACKS/second without time stamps while only (2400/8)/52 = 5.77 ACKS/second can be sent with a time stamp. We know both TCP and SCPS TP are clocked protocols and they use the reception of acknowledgments as an indication that the data has "left the network" so more data can be sent. Therefore, the rate at which the sender receives acknowledgments controls the rate that new data can be sent out. Consequently, fewer acknowledgments per time unit will decrease the amount of data that can be transmitted per unit time.
- At the cost of reducing 12 bytes data for each packet and slowing down the link acknowledgment process, more accurate and current RTT estimates can

be achieved if the time stamps are enabled. This is the real significance of enabling time stamps.

Many people prefer enabling time stamps even at the extra overhead cost. In non-perfect link conditions (i.e., variations in RTT, corruptions and congestion losses), enabling time stamps may help more than hurt. Both TCP and SCPS-TP are tested with timestamps enabled by default for the experiments in this dissertation. The test results for SCPS-TP with timestamps disabled and a detailed description of the impact of timestamp on TCP and SCPS-TP are contained in [6], [7] and [8].

Window Scaling

The 16 bit receiver window size of TCP header limits the largest window that can be used to be 65,536 bytes. TCP performance problems arise with 65 Kbytes receiver window in an "LFN" [25]. The Window Scaling option [24] expands the TCP window size from 16 to 32 bits to permit TCP have more than 64 Kbytes of data outstanding at one time. This was simply done by imposing an implicit scale factor to the advertised window instead of changing TCP header size [16]. This option can increase the maximum outstanding data by powers of two, up to 2^{13}. The Window Scaling options are enabled for both Red Hat Linux 6.1 built-in TCP/IP and SCPS-TP for which the FTP and FP run over individually.

Since SYN segments are always sent reliably, both the sender and the receiver must send the Window Scaling option in their SYN segments to enable window scaling. The passive open end can send the option only if the incoming SYN specifies

it. The scale factor can be different in each direction but should be fixed in each direction when the connection is established.

This option is necessary to fill high capacity packet satellite links that are LFN's [23]. Operating with a larger window is more beneficial in an environment in which data loss is usually caused by link corruption. This makes the sender keep transmitting new data while recovering from packets losses. As one of the techniques for coping with errors, SCPS-TP with the Window Scaling option enabled maintains its throughput in the event of corruption-induced losses.

Similar to RFC 1323 Timestamp option, RFC 1323 Window Scaling option, by default, is enabled for both TCP and SCPS-TP for the experiments in this study. The above strength of the Window Scaling capabilities is not shown up for both protocols in our SGLS test-bed environment but is shown up in our large BDP satellite environment. This is because the test-bed experiments were run over the slow PPP serial link with the maximum rate 57,600 bps and the capacity of less than 670 bytes and the satellite link capacity is much larger than 65 Kbytes. This can be verified from tcptrace traffic statistic report that "adv wind scale" is '0' in SYN segments for test-bed experiments and is non-zero for satellite link tests. The details of window scaling option can be found in [24].

3.1.4 Experiment Assumptions

We made several explicit assumptions about the test configuration and experiment methodology in the previous work. The experimental results we have

used in this dissertation are also based on these assumptions. Here we discuss these assumptions.

3.1.4.1 Number of Test Runs, File Order and File Size

As one of multiple comparison procedures, Fisher's comparison procedure [26] is known as the least significant difference. Based on a *t*-test, Fisher's least significant difference procedure determines that the difference $\overline{y_i} - \overline{y_j}$ is significant if

$$\left|\overline{y_i} - \overline{y_j}\right| \geq t_{\alpha/2,a(n-1)}\sqrt{\frac{2MS_e}{n}}.$$

In which, $\overline{y_i}$ and $\overline{y_j}$ are two sample averages of two treatment groups; α is the significance level of the test, which is mostly defined to be 0.05; MS_e is a pooled estimate of the common variance of the treatment groups; a is the number of treatment groups, n is the number of observations within each group and $a(n\text{-}1)$ are the degrees of freedom of MS_e.

Based on our experience [5], [6], [7], [8], we expect $MS_e \approx 1\,\text{sec}^2$ and the smallest significant mean difference which should be statistically detected is around 1 second, i.e., $\left|\overline{y_i} - \overline{y_j}\right| \approx 1$ second. For our experiments over SGLS test-bed, we have a = 24 configurations for each file within each protocol pair comparison (24=2 Protocol × 2 Channel Rate × 3 BER × 2 Delay). Based on this description, if let α = 0.05, we have

$$1 \geq t_{0.05/2,24(n-1)} \sqrt{\frac{2}{n}}.$$

Let's assume we will need a very huge number of observations to detect the mean difference of 1 second, by a *t*-test, this gives us

$$t_{0.05/2,24(n-1)>120} = 1.96.$$

Therefore, we have

$$1 \geq 1.96 \sqrt{\frac{2}{n}}$$

Solve it, we have $n \geq 7.68$. This indicates that any number which is greater than or equal to 8 can be chosen to be the observation number for our experiments.

Based on the above analysis, let's choose $n = 16$, which is sufficiently large to statistically detect the significant mean difference of 1 second with *Power* ≥ 95%. Most satellite transfers can be thought of a single-attempt trial. The network would have no memory of previous results or chance to optimize based on previous data streams. We consider 16 replicate observations will be representative of these single-shot attempts at data transfers.

In order to prevent systematic bias of file transfer time for each configurations, the data files with size of 1Kbytes, 10Kbytes, 100Kbytes and 1Mbytes will also be arranged randomly for transmission instead of the order from smallest to largest. In total, there will be 2304 (=144×16) runs for the experiment over SGLS test-bed and 576 (=36×16) runs for the experiment over satellite link. The details are provided in Section 3.2.

To avoid cyclic effects associated with random error vector generation, actual file sizes were taken as the prime number nearest the nominal size, as shown below:

1Kbytes	→	997 bytes
10Kbyte	→	10,007 bytes
100Kbyte	→	100,003 bytes
1Mbyte	→	1,000,003 bytes

3.1.4.2 SGLS Test Configuration

The SGLS test configuration can be thought of as a single point-to-point transmission between a ground station and a satellite. There is no external network interaction. It is assumed that the important parameters for this investigation are contained in this link and not in other ground links.

3.1.5 Experiment Procedures

The experiments conducted over SGLS test-bed were performed with the same simulator configuration and software versions. The following subsections discuss the test method and test result analysis techniques.

3.1.5.1 Test Method

As mentioned above, the Expect script files are used to configure and control the simulation process. Basically, the user configures the SGLS for the desired link delay value and BER. The user can set the desired point-to-point link delay from 0 ms through 5000 ms (5 seconds) in our test-bed. The simulated link delays of 0 ms, 3 ms, 120 ms, and 1280 ms in our test-bed correspond to no delay, LEO satellite orbit, GEO satellite orbit, and lunar orbit in realistic environment. If we consider using all the

above four delays levels in our study, our experiment will have 2880 degrees of freedom in the error term. This is a huge number of d.f. We know when an experiment has too many d.f. in its error term, it becomes overcritical. In this case, differences of no practical significance will be found. Considering this effect, in this study, we conducted the experiments with two practical delays of 120 ms and 1280 ms only. The BER can be selected from any of error free, 1E-6 and 1E-5. The Expect script controls the selection of the file size, the number of transmission attempts, and the congestion control mechanism.

3.1.5.2 Data Collection

The Expect scripts produce the transfer time data for each run. These computer-generated data files are then analyzed for data throughput times after being organized manually according to their different protocol and configuration options.

3.1.5.3 Analysis Techniques

We analyze the protocol reported transmission times based on the 16-run averages for each experiment configuration. The file transfer time averages are plotted and compared for interesting configurations using Excel spreadsheets. The experimental data are conducted the Analysis of Variance (ANOVA) and the mean comparisons using the Statistical Analysis System (SAS) based Tukey's Honestly Significant Difference (HSD) procedure [26], [27]. As a typical member of the outside-in class of means separation techniques, Tukey's HSD procedure is generally used for comparing the pairs of treatments and determining which pairs of means are different.

Basically, Tukey's HSD procedure uses a single critical difference

$$q_{\alpha,a,a(n-1)}\sqrt{\frac{MS_e}{n}},$$

that is, two means $\overline{y_i}$ and $\overline{y_j}$ are considered significantly different if

$$\left|\overline{y_i} - \overline{y_j}\right| \geq q_{\alpha,a,a(n-1)}\sqrt{\frac{MS_e}{n}},$$

where q is the α-level critical value of a studentized range distribution of a independent normal random variables with $a(n\text{-}1)$ degrees of freedom, n is the equal size of the experiment group and MS_e is a pooled estimate of the common variance of the treatment groups.

Both of Fisher's least significant difference procedure used in Section 3.1.4.1 and Tukey's HSD procedure are multiple comparison procedures belonging to the "outside-in" class of means-separation techniques. Fish's procedure is known as the least significant difference and is based on a t-test while Tukey's HSD utilizes the studentized range $q_{\alpha,a,a(n-1)}$ which is more conservative. The details of both procedures can be found from [26] and [27].

Alternatively, the significant different pairs of means can also be determined by finding the confidence interval around the mean difference. The confidence interval may be more useful than significance tests in multiple comparisons. Confidence intervals show the degree of uncertainty in each comparison in an easily interpretable way; they make it easier to assess the practical significance of a difference as well as the statistical significance. In our experiments, we determine the

significant different pairs by finding the confidence interval around the difference in the mean file transfer time value. The mean difference is considered to be significant if the confidence interval does not include 0.

In this dissertation, some of different performance graphs obtained using network analysis tools are also used to support our analysis if necessary:

Time Sequence Graph—Time Sequence Graph shows the relationship between the segments and the acknowledgments in terms of time.

Throughput Graph—It shows the average and instantaneous throughput of the connection as a function of time.

Round Trip Time (RTT) Graph—RTT Graph shows the round-trip times for the ACKs as a function of time.

Outstanding Data Graph—It shows the number of packets in flight in the pipe at a particular time.

Segment Size Graph—Segment Size Graph displays how the size of segments varies with respect to time.

The above graphs are obtained by using network analysis tools tcpdump, tcptrace and xplot. The above different graphs can provide us the detailed information about each connection including the elapsed time, size of segments received and transmitted, RTT, throughput and congestion window status. This gives us an view on the relationships between the protocol performance and different network parameters, which cannot be obtained using our previous approach by just comparing the averaged file transmission time [6], [7], [8].

3.2 Experimental Work

As mentioned in Chapter 1, the following two sets of experiments are analyzed in this dissertation:

1. Experiments over SGLS test-bed;
2. Experiments over satellite link.

3.2.1 Experiments over SGLS test-bed

Figure 3.5 outlines test factors and different levels of each factor for experiments over SGLS test-bed. The test conditions include link delay, channel rate, Bit-Error-Rate (BER) and transmission file size, which represent satellite orbit status, channel operating mode, space channel noise and user transmission load individually.

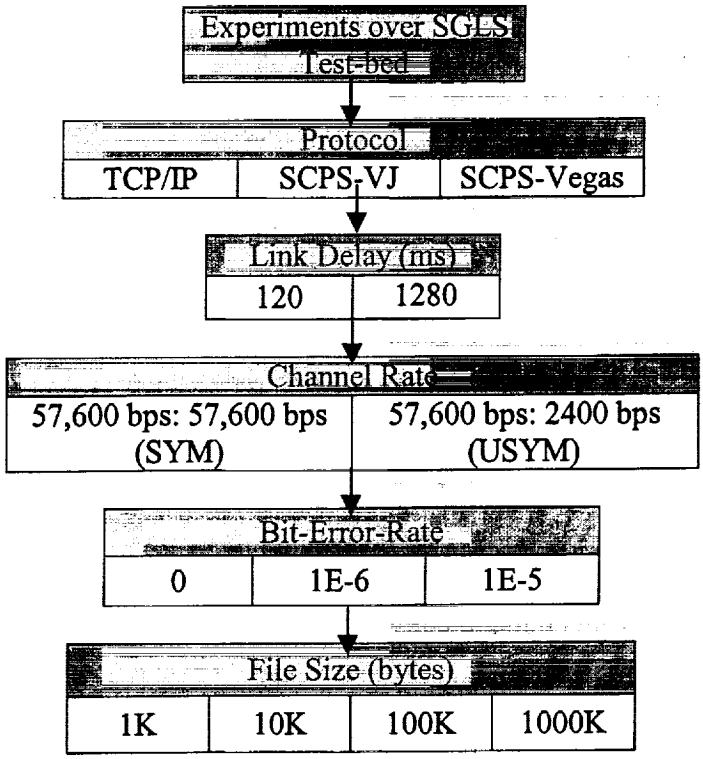

Figure 3.5: Outline showing test factors and different levels of each factor for experiment over SGLS test-bed

The joint of different levels of the above test factors is expected to simulate a sufficiently practical and low BDP space communication environment in which the basic behavior of the protocols can be characterized so that basic questions listed in Chapter 1 can be addressed. By comparing protocol performance between TCP/IP and SCPS and between SCPS-VJ and SCPS-Vegas, we expect to answer the first two questions. By studying the effects of different levels of link delay and the effects of various BER, we expect to address question (3).

From Figure 3.5, we see there will be 144 (=3 Protocol × 2 Delay × 2 Channel Rate × 3 BER × 4 File Size) test configurations for the experiment over SGLS test-bed. Basically, the following three sets of analyses will be done to explore the behavior of protocols by plotting the relationships between the averaged file transfer time (over 16 observations) and the file sizes (1Kbytes, 10Kbytes, 100Kbytes and 1Mbytes). Both the time and the file size are converted into logarithm for an explicit comparison.

(1) Plot and analyze the relationships between the averaged file transfer time and the file sizes for the three protocol options (TCP/IP, SCPS-VJ and SCPS-Vegas) for each of 12 (=2 Channel Rate × 3 BER × 2 Delay) test treatments. So this set includes 12 plots in total.

(2) Plot and analyze the relationships between the averaged file transfer time, the file size for both delay options of 120 ms and 1280 ms with individual

BER for each of TCP/IP, SCPS-VJ and SCPS-Vegas. This includes in total 18 (=3 Protocol × 2 Channel Rate × 3 BER) plots.

(3) Plot and analyze the relationships between the averaged file transfer time and the file sizes for all three BERs with each of two delays for each of TCP/IP, SCPS-VJ and SCPS-Vegas. This will have 12 (=3 Protocol × 2 Channel Rate × 2 Delay) plots.

For the above three sets of analyses, set (1) is intended as a straight comparison of the performance of TCP/IP with SCPS and to compare the performance between two control modes of SCPS itself under the identical transmission conditions. Analysis set (2) is intended to investigate how each protocol behaves differently when a much longer link delay is involved with the increase of the file size under different combination of channel rates and BERs. The objective for above analysis set (3) is to explore how each protocol behaves differently for different BERs along with the change of the file size under different test conditions of channel rate and delay.

For set (1), the experimental data will be classified into the following two sets for protocol performance comparison using the SAS procedures:

(1) TCP/IP based data versus SCPS-VJ based data;

(2) SCPS-VJ based data versus SCPS-Vegas based data.

The above classification is based on the relationship between the protocols and two control modes of SCPS protocol. Like standard Van Jacobson (VJ) congestion control based TCP, SCPS-TP makes a default assumption regarding the

source of loss in the absence of any explicit information. TCP's default assumption is that all loss is caused by congestion; however, SCPS-TP's default parameter can be set either to SCPS-VJ or to SCPS-Vegas by an application based on different assumptions of the source of packet loss. In a realistic satellite environment, where network bandwidth is primarily managed on private links, link congestion is unlikely and it is reasonable for SCPS-TP to assume by default that any loss is due to errors. The congestion control philosophy for SCPS-VJ is the same as that for TCP which is to assume that all data loss (regardless of bit error loss or link congestion loss) is caused by the link congestion while the philosophy for SCPS-Vegas distinguishes the loss caused by bit error and congestion. Another words, TCP and SCPS-VJ consider the bit error loss as link congestion loss while SCPS-Vegas treats bit error just as bit error. Based on the above different assumptions, for our experiments over SGLS test-bed and realistic satellite link where the bit error dominates the data loss, VJ based TCP and SCPS-VJ consider high BER caused data loss as congestion loss and thus reduce the congestion window and further slow down the transmission while SCPS-Vegas might keep its throughput unchanged in the case of frequent data loss caused by high BER. This will definitely affect the protocol performance. Based on the above description, the comparison set (1) is intended to provide an intuitive performance comparison between VJ based protocols TCP and SCPS under the same assumption that all data loss is caused by congestion. The objective for the comparison set (2) is to see how SCPS performs differently under the different

assumptions that data loss is caused either by link congestion or by bit error corruption.

Based on different features implemented for each protocol described in Chapter 2 and the above description of test factors, Table 3.1 provides a qualitative expectation of space channel effects on three protocols. These effects on protocol performance are studied in detail in Section 4.2 and Section 4.3.

Table 3.1: Expected qualitative effects of space channel conditions on protocols

	TCP/IP	SCPS-VJ	SCPS-Vegas
Link Delay	Highly Sensitive	Moderately Sensitive	Slightly Sensitive
Channel Rate	Moderately Sensitive	Slightly Sensitive	Slightly Sensitive
Bit-Error-Rate	Highly Sensitive	Moderately Sensitive	Slightly Sensitive

Corresponding to basic questions listed in Chapter 1 and the above three sets of analyses, the following sets of null hypotheses may be tested using the HSD procedure:

For analysis set (1):

Hypothesis Set 1: TCP/IP and SCPS-VJ have equal file transfer time means for each of the same transmission conditions of link delay, BER and file size with symmetric channel rate.

Hypothesis Set 2: SCPS-VJ and SCPS-Vegas have equal file transfer time means for each of the same transmission conditions of link delay, BER and file size with symmetric channel rate.

Hypothesis Set 3: TCP/IP and SCPS-VJ have equal file transfer time means for each of the same transmission conditions of link delay, BER and file size with asymmetric channel rate.

Hypothesis Set 4: SCPS-VJ and SCPS-Vegas have equal file transfer time means for each of the same transmission conditions of link delay, BER and file size with asymmetric channel rate.

The test results for the above hypotheses Set 1 and Set 2 can be found in Section 4.1.1 and the results for Set 3 and Set 4 are in Section 4.1.2.

For analysis set (2):

Hypothesis Set 1: TCP/IP has equal file transfer time means with Delay=120 ms and Delay=1280 ms for each of the same transmission conditions of channel rate, BER and file size.

Hypothesis Set 2: SCPS-VJ has equal file transfer time means with Delay=120 ms and Delay=1280 ms for each of the same transmission conditions of channel rate, BER and file size.

Hypothesis Set 3: SCPS-Vegas has equal file transfer time means with Delay=120 ms and Delay=1280 ms for each of the same transmission conditions of channel rate, BER and file size.

Sections 4.2.1, 4.2.2 and 4.2.3 provide the test results for each of the above three sets of test hypotheses.

For analysis set (3):

Hypothesis Set 1: TCP/IP has equal file transfer time means with BER=0 and BER=1E-6 for each of the same transmission conditions of channel rate, link delay and file size.

Hypothesis Set 2: TCP/IP has equal file transfer time means with BER=1E-6 and BER=1E-5 for each of the same transmission conditions of channel rate, link delay and file size.

Hypothesis Set 3: SCPS-VJ has equal file transfer time means with BER=0 and BER=1E-6 for each of the same transmission conditions of channel rate, link delay and file size.

Hypothesis Set 4: SCPS-VJ has equal file transfer time means with BER=1E-6 and BER=1E-5 for each of the same transmission conditions of channel rate, link delay and file size.

Hypothesis Set 5: SCPS-Vegas has equal file transfer time means with BER=0 and BER=1E-6 for each of the same transmission conditions of channel rate, link delay and file size.

Hypothesis Set 6: SCPS-Vegas has equal file transfer time means with BER=1E-6 and BER=1E-5 for each of the same transmission conditions of channel rate, link delay and file size.

Section 4.3.1 will provide the test results for Set 1 and Set 2; Section 4.3.2 will have results for Set 3 and Set 4 and Section 4.3.3 for Set 5 and Set 6.

3.2.2 Experiments over Satellite Link

Figure 3.6 outlines test factors and different levels of each factor for experiment over satellite link. Similar to the experiments over the SGLS test-bed, by comparing protocol performance between TCP/IP and SCPS, we expect to answer question (1) and question (2) listed in Chapter 1. We also expect to address question (4) by comparing the protocol performance between SGLS test-bed and live satellite link. Besides this, the tests over realistic satellite link are also expected to bring more benefits: (1) Extending performance to cover large BDP region as well; (2) Improving the SGLS test-bed based on the analysis of test results

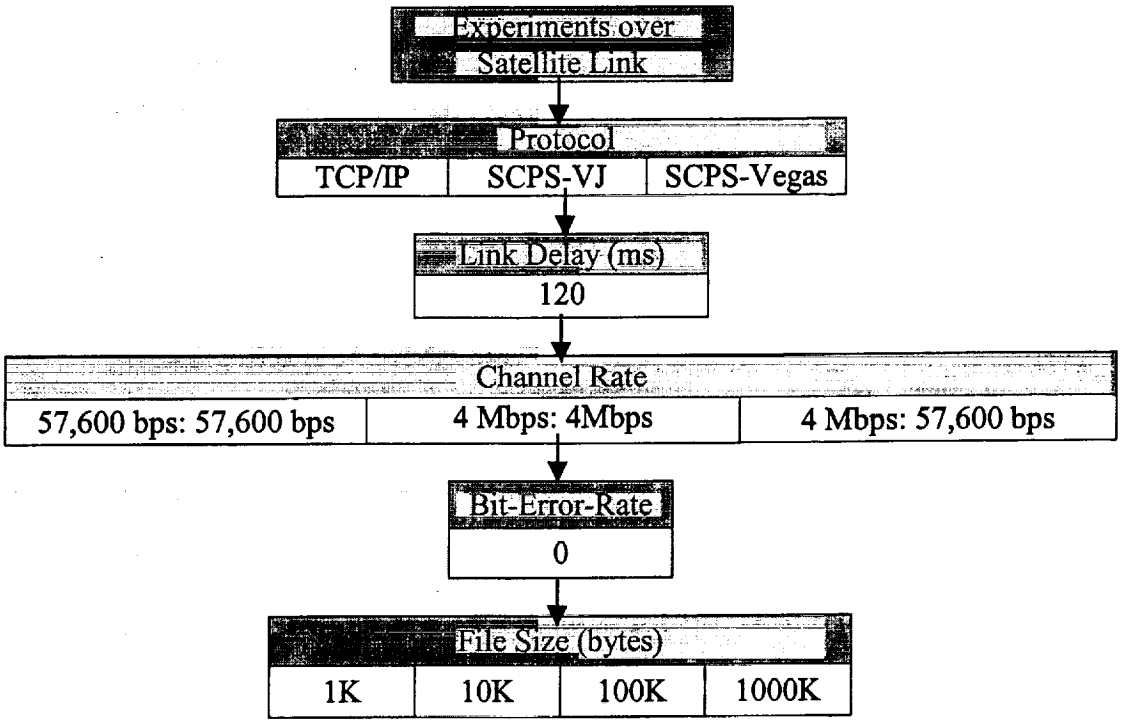

Figure 3.6: Outline showing test factors and different levels of each factor for experiment over satellite link

By comparing Figure 3.5 and Figure 3.6, we see there are three differences between them:

(1) Tests over satellite link are done only with the delay of 120 ms;

(2) Tests over satellite link are done only with error free link;

(3) Only rate of 57,600 bps:57,600 bps is identical between them.

All the above three differences are due to the restriction of available satellite link conditions. Difference (1) is due to the fact that the available geostationary satellite has the link delay fixed at 120 ms. For difference (2), the condition of nonzero BERs could not be obtained although many efforts were made by both Naval Research Lab (NRL) and Infinite Global Infrastructure (IGI) which both are our experiment cooperators. The tests over nonzero BERs are left to be future work when the conditions are available. For difference (3), the tests with rate of 4 Mbps:4 Mbps and rate of 4 Mbps:57,600 bps are done to extend the performance analyses to cover large BDP region. With the rate of 57,600 bps:57,600 bps, we aim to validate the SGLS test-bed performance by comparing the in-lab results with the actual satellite channel results under the conditions of slow symmetric channel rate. This is expected to help us improve our SGLS test-bed based on the comparison results. The proposed tests with the rates of 57,600 bps:2400 bps and 4 Mbps:9600 bps could not be conducted since the slowest satellite link rate available is around 57,600 bps.

From Figure 3.6, we see the experiment over satellite link will have 36 (=3 Protocol × 1 Delay × 3 Channel Rate × 1 BER × 4 File Size) configurations.

Considering that the tests over satellite link are conducted under different conditions, satellite channel results will be analyzed in the following two sets:

(1) Plot and analyze the relationship between the averaged file transfer time and the file size for three protocol options (TCP/IP, SCPS-VJ and SCPS-Vegas) for each of 3 (=3 Channel Rate × 1 BER × 1 Delay) configurations.

(2) Plot and analyze the relationship between the averaged file transfer time and the file size with rate of 57,600 bps:57,600 bps, BER=0 and delay of 120 ms for both SGLS test-bed and satellite link for each of 3 (=1 Channel Rate × 3 Protocol × 1 BER × 1 Delay) configurations.

From the first way, we expect to see the performance differences among different protocols under the same transmission condition over satellite link. The second way is intended to explore the performance differences and/or similarities between SGLS and satellite link for each of three protocols and thus, to validate the SGLS test-bed performance.

Similar to the experiments over test-bed, the satellite link protocol performance analysis set (1) will be done based on analyzing the following two sets of relationships using the SAS based procedure:

(1) TCP/IP based data versus SCPS-VJ based data;

(2) SCPS-VJ based data versus SCPS-Vegas based data.

The testable sets of null hypotheses corresponding to two sets of protocol performance analyses are listed below:

For analysis set (1):

Hypothesis Set 1: TCP/IP and SCPS-VJ have equal file transfer time means for each of three channel rates and each of four file sizes with BER=0 and Delay=120 ms.

Hypothesis Set 2: SCPS-VJ and SCPS-Vegas have equal file transfer time means for each of three channel rates and each of four file sizes with BER=0 and Delay=120 ms.

Section 5.1 will give the test results for the above two sets of test.

For analysis set (2):

Hypothesis Set 1: TCP/IP has equal file transfer time means for tests over SGLS test-bed and satellite link with each of four file sizes, channel rate 57,600 bps:57,600 bps, BER=0 and Delay=120 ms.

Hypothesis Set 2: SCPS-VJ has equal file transfer time means for tests over SGLS test-bed and satellite link with each of four file sizes, channel rate 57,600 bps:57,600 bps, BER=0 and Delay=120 ms.

Hypothesis Set 3: SCPS-Vegas has equal file transfer time means for tests over SGLS test-bed and satellite link with each of four file sizes, channel rate 57,600 bps:57,600 bps, BER=0 and Delay=120 ms.

The test results for the above three sets of test hypotheses are provided in Section 5.2.

Necessary analysis of variance is conducted to compare the protocol performance and analyze the effects of link delay and BER on each protocol

performance to test the above hypotheses. Additionally, linear regression models are built for experiments over the SGLS test-bed to reflect the relationships between protocols' file transfer time and transmission conditions. Regression models are built for the regression of logarithmic file transfer time on file size, delay, BER and their interactions for each of joint conditions of protocol and channel rate. Those models are built when we study the effects of BER on the protocol performance in Section 4.3. Linear regression models for the regression of satellite link time on SGLS test-bed time are also built for the study of the protocol performance over satellite link in Chapter 5. Building a conventional response surface for the experimental data may not be successful. We know quadratic response surfaces are a multivariate Taylor series expansion of the regression surface. Consequently, the methodology requires at least three distinct values for each explanatory variable (this is a necessary condition but is not sufficient by itself). Two quantitative explanatory variables in our experiments, Bit-Error-Rate and File Size, having three distinct levels in which Bit-Error-Rate contains 0, 1E-6 and 1E-5 and File-Sizes were nominally 1K, 10K, 100K and 1000K. As can be seen, the BER spans six orders of magnitude and the file size span four orders of magnitude. An experimental region of this size may create serious problems for polynomial approximation. These problems will show up in the formal lack-of-fit test: the tests statistic is enormous. In this case, the response surface may come nowhere near the observed means for most combinations observed. This can be understood from the way that the data indicate that a 2-order polynomial surface

cannot bend itself into the required shape and thus, reasonable response surfaces cannot be built for the analysis.

Chapter 4 and Chapter 5 present the detailed analysis of the experiment over the SGLS test-bed and the experiment over satellite link respectively.

4 BEHAVIOR OF TCP/IP AND SCPS OVER THE SGLS TEST-BED

This chapter analyzes the behavior of TCP and SCPS (SCPS-VJ and SCPS-Vegas) running different congestion control modes over the SGLS test-bed by plotting the averaged file transfer time of each file size for different experiment runs. As mentioned in Section 3.2.1, the following three sets of plots will be examined:

(1) Plot the relationships between the averaged file transfer time and the file sizes for the three protocol options (TCP/IP, SCPS-VJ and SCPS-Vegas) for each of the twelve test configurations.

(2) Plot the relationships between the averaged file transfer time, the file size for both delay options of 120 ms and 1280 ms with individual BER for each of TCP/IP, SCPS-VJ and SCPS-Vegas control options.

(3) Plot the relationships between the averaged file transfer time, and the file size for all three BERs with each of two delays for each of TCP/IP, SCPS-VJ and SCPS-Vegas.

Set (1) is intended as a straight comparison of the performance of TCP/IP with SCPS and to compare the performance between two control modes of SCPS itself under the identical transmission conditions. Analysis set (2) is intended to investigate how each protocol behaves differently when a much longer link delay is involved with the increase of the file size under different combination of channel rates and BERs. The objective for above analysis set (3) is to explore how each protocol behaves differently for different BERs along with the change of the file size under different test conditions of channel rate and delay.

When we study the BER effects on protocol performance in set (3), linear regression models for the regression of file transfer time on the transmission conditions are also built to reflect the relationships between the response time and experimental factors in our experiments.

The above three sets of plots are intended to compare the performance of TCP/IP and SCPS and to investigate the effects of delay and BERs on the protocol performance. When we examine the above three sets of plots, the mean comparison using the SAS based HSD procedure is also provided in the form of a table. Each comparison table contains corresponding file size, mean times for each comparison pair, mean difference in seconds, 95% confidence interval and mean difference in percentage. The mean difference is considered to be statistically significant if the confidence interval does not include 0. Comparisons significant at the experiment wise error rate 0.05 level are indicated by "*" following confidence limits. The mean difference in percentage is calculated only for each pair which has statistically significant difference. The performance comparison tables are used to support our analyses of plots by providing a quantitative difference and a qualitative result for each mean comparison pair.

Sections 4.1, 4.2 and 4.3 concentrate on each of the above three sets of analyses. The analysis will be primarily supported using the SAS based HSD procedure. Appendices A, B and C provide the mean and standard deviation of file transfer time of 16 observations for each experimental run, which are used for the above plots.

4.1 Comparing the Performance of TCP and SCPS

The goal of the performance comparison between protocols is to see which protocol has better performance under various transmission conditions. As mentioned in Chapter 3, based on the relationship between the protocols and the relationship between two control modes of SCPS protocol, the analysis using SAS based HSD procedure will be done for each of the following protocol comparison sets:

(1) TCP/IP versus SCPS-VJ;

(2) SCPS-VJ versus SCPS-Vegas.

Each of the above two comparison sets has 24 configurations (=2 Protocol × 2 Channel Rate × 3 BER × 2 Delay) for each file size. The number of observations is 384 (=16 × 24) since there are 16 observations for each treatment. Two sets of comparisons will be done for each of two protocol comparison sets based on different channel rates: symmetric rate (115,200 bps:115,200 bps) and asymmetric rate (115,200 bps:2400 bps). Each set of comparison will be made for each BER with delays of 120 ms and 1280 ms.

4.1.1 Comparison with Symmetric Channel Rate of 115,200 bps:115,200 bps

The simulated channel rate of 115,200 bps:115,200 bps over test-bed is considered to simulate slow symmetric satellite channel.

4.1.1.1 BER=0

Tests with BER=0 are expected to predict the protocol performance over error free satellite link.

Delay of 120 ms

The simulated channel delay of 120 ms is considered to correspond to GEO satellite orbit. Figure 4.1 compares the transfer time of each file size for all TCP, SCPS-VJ and SCPS-Vegas protocols with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms. Note both file transfer time and the file size are plotted in logarithm for an explicit comparison. The data table at the bottom contains the averaged logarithm time for each of 12 combinations of the protocol and file size.

Table 4.1 provides the corresponding comparisons of means for two protocol comparison sets using Tukey's HSD procedure.

When we look at the plot in Figure 4.1, we see the means for all file size among three protocols are bound together except for 1K file where TCP/IP takes a bit more time than both SCPS-VJ and SCPS-Vegas do. The actual mean difference is about 0.068 second (=0.145-(0.076+0.078)/2). Although there is a slight difference in this case, the result of the comparison of means in Table 4.1 shows there is no statistically significant difference for all eight pairs of means of four files between three protocols. Thus, we may conclude that three protocols perform essentially the same for the slow symmetric, error-free channel with 120 ms delay.

Delay of 120 ms

Figure 4.2 plots the file transfer time for all protocols with delay of 1280 ms. From Figure 4.2, we see SCPS-Vegas jumps over both TCP/IP and SCPS-VJ at the points 10K and 100K files while TCP/IP and SCPS-VJ keep closely during the whole course of four file transmission.

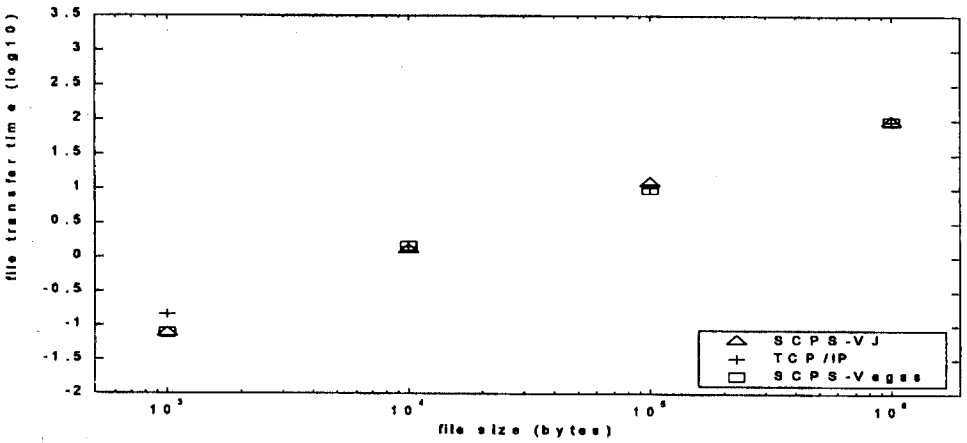

Figure 4.1: File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

Table 4.1: Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit		Means Difference (%)
1K	0.145	0.076	0.0685	-0.6185	0.7555	--
10K	1.420	1.278	0.142	-5.690	5.974	--
100K	10.456	12.321	-1.865	-8.933	5.203	--
1000K	96.875	94.823	2.052	-5.055	19.159	--
File Size	**SCPS-Vegas Mean (secs)**	**SCPS-VJ Mean (secs)**	**Means Difference (secs)**	**95% Confidence Limit**		**Means Difference (%)**
1K	0.078	0.076	0.0017	-0.6813	0.6847	--
10K	1.442	1.278	0.1639	-0.9487	1.2766	--
100K	10.190	12.321	-2.131	-6.009	1.746	--
1000K	95.351	94.823	0.528	-8.448	9.504	--

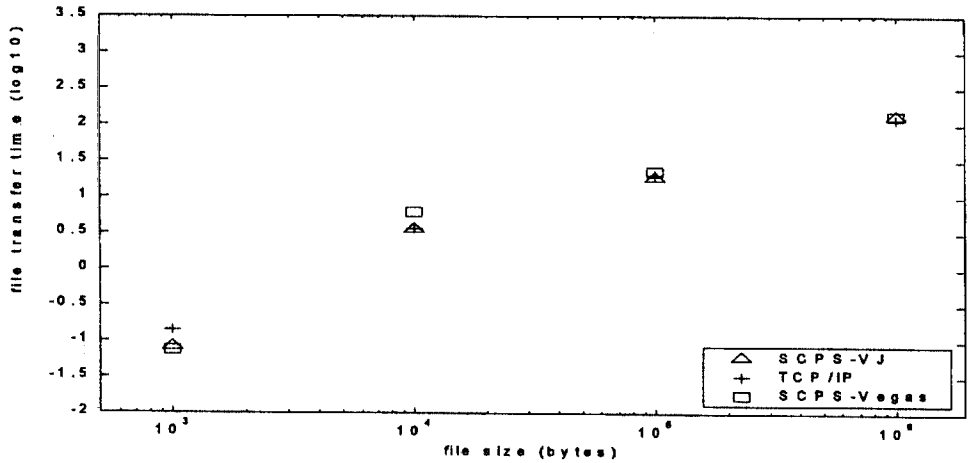

Figure 4.2: File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=1280 ms

Table 4.2: Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=1280 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.137	0.082	0.0550	-0.6320	0.7421		--
10K	3.634	3.612	0.021	-5.811	5.853		--
100K	20.244	18.729	1.515	-5.553	8.584		--
1000K	120.438	137.043	-16.605	-33.712	0.502		--
File Size	**TCP Mean (secs)**	**SCPS-VJ Mean (secs)**	**Means Difference (secs)**	**95% Confidence Limit**			**Means Difference (%)**
1K	0.073	0.082	-0.0091	-0.6921	0.6739		--
10K	6.074	3.612	2.4615	1.3488	3.5741	*	68.2%
100K	22.675	18.729	3.947	0.069	7.824	*	21.1%
1000K	135.183	137.043	-1.860	-10.836	7.116		--

The HSD procedure in Table 4.2 indicates that the only two points whose means are significantly different: 10K and 100K points between SCPS-Vegas and SCPS-VJ. This supports the observations in Figure 4.2.

By comparing the above two plots with BER=0, we may conclude for the comparing the protocol performance over error free link:

- There is no significant difference in the performance of the protocols for all four file size with a delay of 120 ms.
- The increase of link delay time for 120 ms to 1280 ms does not affect the relationship between three protocols for a very small file (1K) and a relatively large file (1000K) but causes statistically significant differences for 10K and 100K files between SCPS-VJ and SCPS-Vegas. By checking all 16 observations for both SCPS-VJ and SCPS-Vegas, these differences were not caused by particular exceptional runs. These differences actually come from the fundamental behavior difference between VJ's traditional Slow Start mechanism and Vegas's modified Slow Start mechanism. For the transmission of a small file such as 10K or 100K, most of the transmission work is done during the initial Slow Start phase. As we discussed in Section 2.1.4, in order to detect and prevent congestion occurring during Slow Start phase, Vegas modifies traditional Slow Start from exponential growth of throughput for every RTT to exponential growth for every other RTT. In between, the congestion window stays fixed so a valid comparison of the expected rate and actual rates can be made. When the actual rate falls below the expected rate by

the equivalent of one router buffer, Vegas changes from exponentially increasing Slow Start phase to linearly increasing Congestion Avoidance phase. For transmitting a relative small file such as 10K or 100K, the slow exponential growth of the congestion window only for every other RTT definitely decreases Vegas's throughput. This is what we see here for the transfer of 10K and 100K files. When we have a very large file such as 1000K, most of the transmission is done during the Congestion Avoidance phase. Vegas's "proactive" Congestion Avoidance mechanism detects incipient congestion and avoids packet losses and thus compensates earlier slow rate transmission during Slow Start phase. This is what happens for 1000K where no statistical performance difference can be seen. Although there is 68.2% Mean Difference(%) for 10K and 21.1% Mean Difference(%) for 100K, they may not be practically significant since both the mean times and the file sizes are actually very small.

- We might expect that, along the increase of the BER, Vegas might perform better than SCPS-VJ for 1000K file while it will still perform behind for 10K file with Mean Difference(%) getting smaller and smaller. This is because a very high BER causes very frequent packet losses for SCPS-VJ (actually also for TCP/IP since both run the same congestion control algorithms) and thus reduces its throughput while Vegas's modified mechanisms prevent those losses and keep its consistent transmission.

- The delay increase does not change the relationships between TCP/IP and SCPS-VJ and their performance keeps no statistical significantly difference.
- Based on the above observation, we may expect that decreasing link delay to around 3 ms (i.e., extending it to LEO link) would not make the protocol performance and the performance relationship too much different since the link delay difference between 3 ms to 120 ms is practically trivial compared with frame clock out time.

4.1.1.2 BER=1E-6

The BER of 1E-6 is not expected to make serious data corruption and does not cause too much retransmissions. Thus, the throughput is not expected to be decreased seriously. This should be clearer for transmitting small files.

Delay of 120 ms

Figure 4.3 displays the relationships among protocols with channel rate of 115,200 bps:115,200 bps, BER=1E-6 and Delay=120 ms.

We note the relationships among protocols are very similar to that from the comparison with BER=0 in Figure 4.1. An intuitive idea is that all three protocols perform similarly. This can be verified by looking at the HSD procedure in Table 4.3, which shows there is no pair having significant performance difference.

Delay of 1280 ms

Protocol relationship for delay of 1280 ms is plotted in Figure 4.4, which shows that the relationship is similar to that in Figure 4.2 except that three protocols have separation at the point of 1000K file.

When we look at the comparison of means in Table 4.4, we see all three protocols perform significantly different for 1000K file size with that SCPS-Vegas's time is the least and TCP/IP's is the most.

As we expected in Section 4.1.1.1, Vegas performs better than SCPS-VJ for 1000K file while it is still behind for 10K file with a smaller Mean Difference(%). This phenomenon is expected to be clearer when the BER is increase to 1E-5. We also realize that the variances in the file transfer times are getting larger with the increase of BER from 0 to 1E-6.

Based on the above analyses for both delays, we may have the following conclusions for the protocol comparison with BER of 1E-6:

- With the delay of 120 ms, the change from error free to BER=1E-6 does not significantly change the performance of all protocols and their relationships, and the protocols still perform similarly.
- The combinations of BER=1E-6 and Delay =1280 ms make all three protocols perform significantly different each other for a large file 1000K with Mean Differences(%) larger than 15%. We may consider that three protocols are practically different for transmitting 1000K file with BER=1ER-6 and Delay=1280 ms. This can be understood that the file of 1000 Kbytes is large enough to lead the protocols into the steady state with the effects of error corruption and longer link delay so that their performance difference is shown up.

4.1.1.3 BER=1E-5

BER=1E-5 is considered to be a relatively high error rate for ground internet channels. But it is still within the space communication specifications of NASA.

Delay of 120 ms

Figure 4.5 plots the performance of three protocols for channel rate of 115,200 bps:115,200 bps, BER=1E-5 and Delay=120 ms. We see here all files averaged transfer time are much longer than that in Figure 4.1 and Figure 4.3. But their relationship is still similar to previous two except that, for 1000K file, SCPS-Vegas taking the least time performs significantly different from other two which almost have no performance difference. Table 4.5 supports our observation.

Delay of 1280 ms

Figure 4.6 shows the situation when protocols are used to transmit file with BER=1E-5 and Delay=1280 ms. Three protocols all perform differently each other for relatively large files 100K and 1000K, especially for 1000K file between TCP/IP and SCPS-Vegas. As we expected in Section 4.1.1.1, Vegas performs much better than SCPS-VJ for 1000K (and 100K) file while it is still behind for 10K file but with a smallest Mean Difference(%) 37.7% comparing with 58.9% and 68.2% for 1E-6 and error free as we saw. When we look at corresponding variances for both BER=1E-6 and BER=1E-5 at the Appendices, we see all variances are getting larger for both delays along with the increase of BER from 0 through 1E-6 to 1E-5.

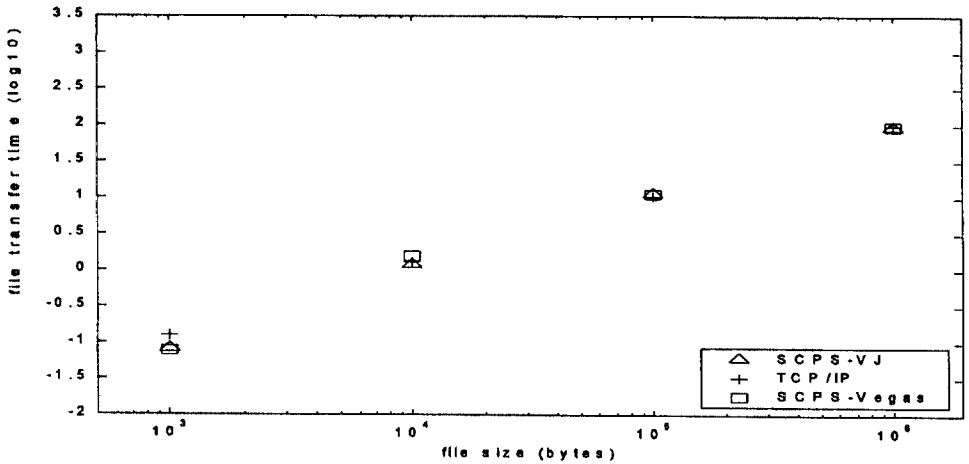

Figure 4.3: File transfer time for TCP/IP, SCPS-VJ and SCSP-Vegas with channel rate of 115,200 bps:115,200 bps, BER=1E-6 and Delay=120 ms

Table 4.3: Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=1E-6 and Delay=120 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit		Means Difference (%)
1K	0.126	0.082	0.0443	-0.6427	0.7313	--
10K	1.314	1.219	0.094	-5.738	5.926	--
100K	10.544	11.585	-1.041	-8.109	6.028	--
1000K	100.519	97.910	2.608	-14.499	19.716	--
File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit		Means Difference (%)
1K	0.078	0.082	-0.0034	-0.6864	0.796	--
10K	1.568	1.219	0.3481	-0.7645	0.608	--
100K	11.052	11.585	-0.533	-4.411	0.44	--
1000K	97.402	97.910	-0.508	-9.484	8.0	--

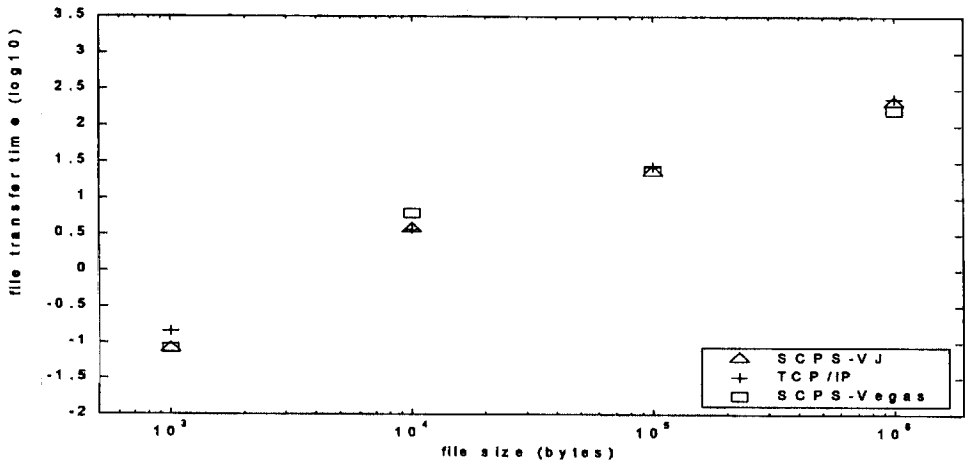

Figure 4.4: File transfer time for TCP/IP, SCPS-VJ and SCSP-Vegas with channel rate of 115,200 bps:115,200 bps, BER=1E-6 and Delay=1280 ms

Table 4.4: Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=1E-6 and Delay=1280 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.147	0.082	0.0647	-0.6223	0.7517		--
10K	3.697	3.829	-0.132	-5.964	5.700		--
100K	27.699	23.929	3.740	-3.328	10.808		--
1000K	239.125	207.292	31.833	14.725	48.940	*	15.4%
File Size	**Vegas Mean (secs)**	**SCPS-VJ Mean (secs)**	**Means Difference (secs)**	**95% Confidence Limit**			**Means Difference (%)**
1K	0.084	0.082	0.0018	-0.6812	0.6848		--
10K	6.086	3.829	2.2569	1.1443	3.3695	*	58.9%
100K	24.102	23.929	0.173	-3.704	4.051		--
1000K	162.009	207.292	-45.283	-54.259	-36.307	*	-21.9%

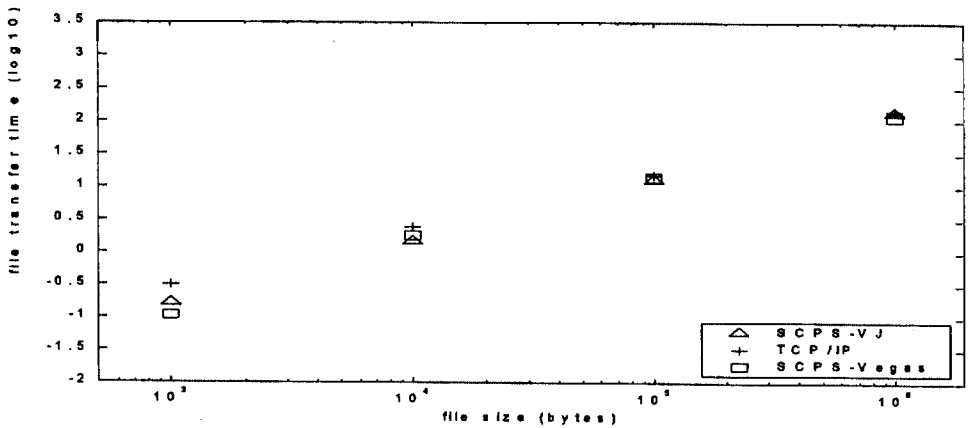

Figure 4.5: File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=1E-5 and Delay=120 ms

Table 4.5: Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=1E-5 and Delay=120 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.312	0.167	0.1445	-0.5425	0.8315		--
10K	2.450	1.484	0.966	-4.866	6.798		--
100K	15.088	13.023	2.064	-5.004	9.133		--
1000K	146.063	139.901	6.162	-10.945	23.269		--
File Size	**SCPS-Vegas Mean (secs)**	**SCPS-VJ Mean (secs)**	**Means Difference (secs)**	**95% Confidence Limit**			**Means Difference (%)**
1K	0.109	0.167	-0.0589	-0.7419	0.6241		--
10K	1.789	1.484	0.3055	-0.8071	1.4182		--
100K	14.066	13.023	0.983	-2.894	4.861		--
1000K	117.240	139.901	-22.660	-31.636	-13.684	*	-16.2%

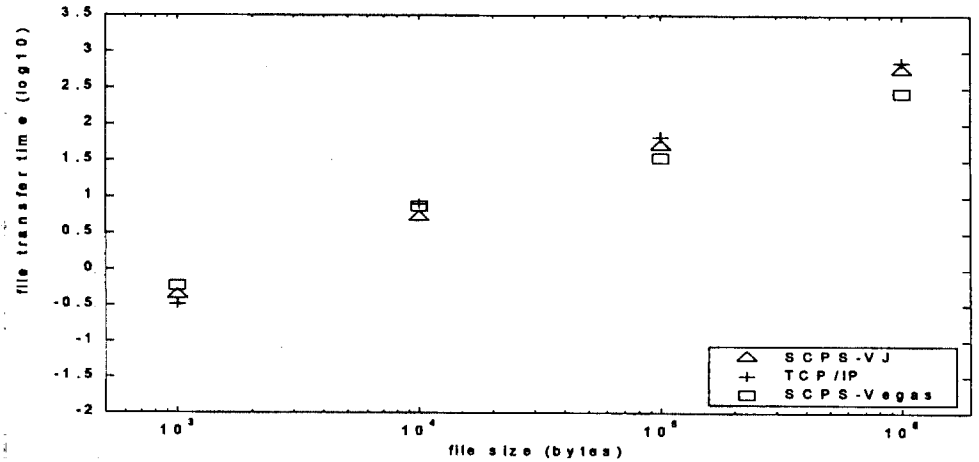

Figure 4.6: File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=1E-5 and Delay=1280 ms

Table 4.6: Comparison of means for channel rate of 115,200 bps:115,200 bps, BER=1E-5 and Delay=1280 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	.328	0.435	-0.1071	-0.7941	0..5799		--
10K	8.097	5.340	2.757	-3.075	8.588		--
100K	66.919	51.649	15.270	8.201	22.338	*	29.6%
1000K	717.688	567.325	150.363	133.255	167.470	*	26.5%
File Size	SCPS-Vegas Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.603	0.435	0.1672	-0.5158	0.8503		--
10K	7.351	5.340	2.0104	0.8978	3.1231	*	37.7%
100K	33.944	51.649	-17.705	-21.582	-13.828	*	-34.3%
1000K	267.870	567.325	-299.455	-308.431	-290.479	*	-52.8%

From Table 4.6 we see that, for 1000K file, there exists a 300 seconds difference between SCPS-VJ and SCPS-Vegas and a 150 seconds difference between SCPS-VJ and TCP/IP. Mean Differences(%) for 1000K are large so they should be considered practically different. Since all Mean Differences among protocols are very large so they should be considered really significantly different. Based on the above analyses for symmetric channel rate, we found that a very long link delay and a very high BER really degrade the protocol performance. This is reasonable and is also expected. The results show that, under this transmission condition, TCP/IP performs poorly and SCPS-Vegas has the best performance.

4.1.2 Comparison with Asymmetric Channel Rate of 115,200 bps:2400 bps

4.1.2.1 BER=0

Delay of 120 ms

Figure 4.7 plots the protocol performance for the channel rate of 115,200bps:2400 bps, BER=0 and Delay =120 ms. Comparing to the plot for symmetric rate, BER=0 and Delay=120 ms in Figure 4.1, we see the averaged file transfer time for all protocols seem to be longer for the corresponding files and protocols are basically bound together. We can also see a bit separation of TCP/IP from other two for 1K file and a little separation of SCPS-Vegas from other two for 10K file. These are not expected to indicate significant performance difference. This can be verified by looking at the HSD procedure output in Table 4.7 which shows no pairs being significantly different. We may conclude, for the simulated error free geostatioanary satellite link, the change of uplink rate from 115,200 bps to 2400 bps

decreases the protocol throughput but does not significantly change the relationship among them.

Delay of 1280 ms

When the delay time is increased to 1280 ms, the statistically significant differences are displayed in the performance means for files 10K, 100K and 1000K as shown in Figure 4.8, we see that SCPS-VJ's performance is statistically different from other two around 10K and 100K file and is different from TCP/IP for 1000K file where no difference showed from SCPS-Vegas. So the comparison conclusion is that the performance means of SCPS-VJ have statistically significant difference from that of TCP/IP for slow asymmetric channel with error free and delay of 1280 ms. Both Mean Differences between TCP/IP and SCPS-VJ are large for 100K and 1000K files while SCPS-Vegas and SCPS-VJ shows no significant differences. We may be seeing real protocol differences between TCP/IP and SCPS-VJ and no differences between SCPS-Vegas and SCPS-VJ. Similar to symmetric channel rate, the performance difference for 10K file between SCPS-Vegas and SCPS-VJ is caused by the fundamental difference between two Slow Start mechanisms of VJ and Vegas as discussed in Section 4.1.1.1. We might also expect that, along the increase of the BER, SCPS-Vegas will show the highest throughput for 1000K file and will perform behind than SCPS-VJ for 10K file.

Table 4.8 shows the details of the comparison of means.

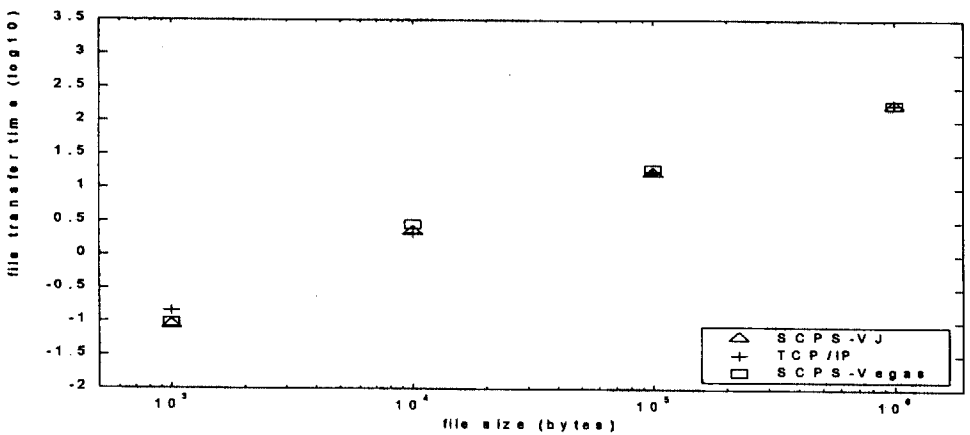

Figure 4.7: File transfer time for TCP/IP, SCPS-VJ and SCSP-Vegas with channel rate of 115,200 bps:2400 bps, BER=0 and Delay=120 ms

Table 4.7: Comparison of means for channel rate of 115,200 bps:2400 bps, BER=0 and Delay=120 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit		Means Difference (%)
1K	0.145	0.089	0.0566	-0.6304	0.7436	--
10K	2.218	2.255	-0.127	-5.958	5.705	--
100K	17.269	17.022	0.247	-6.822	7.315	--
1000K	173.500	174.482	-0.982	-18.089	16.125	--
File Size	SCPS-Vegas Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit		Means Difference (%)
1K	0.095	0.089	0.0065	-0.6766	0.6895	--
10K	2.814	2.255	0.5590	-0.5536	1.6717	--
100K	18.976	17.022	1.954	-1.923	5.831	--
1000K	177.758	174.482	3.276	-5.700	12.252	--

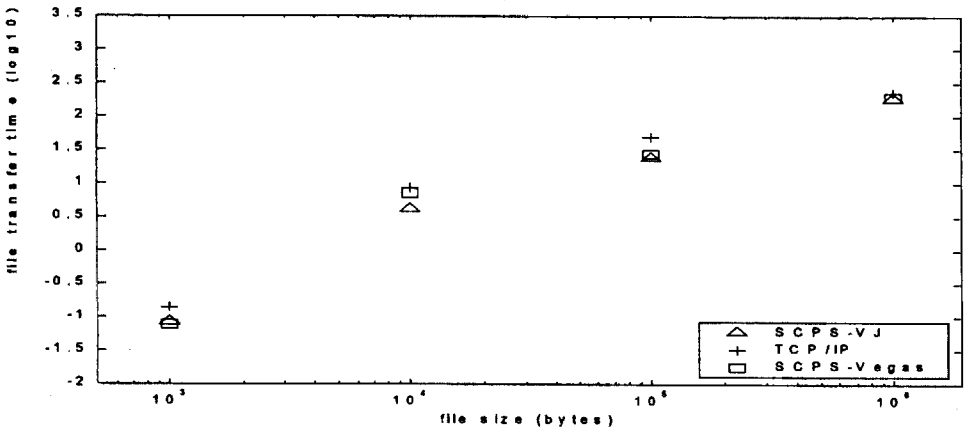

Figure 4.8: File transfer time for TCP/IP, SCPS-VJ and SCSP-Vegas with channel rate of 115,200 bps:2400 bps, BER=0 and Delay=1280 ms

Table 4.8: Comparison of means for channel rate of 115,200 bps:2400 bps, BER=0 and Delay=1280 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.140	0.086	0.0539	-0.6331	0.7410		--
10K	8.689	4.231	4.458	-1.374	10.290		--
100K	50.156	24.250	25.952	18.883	33.020	*	107%
1000K	222.428	190.625	31.812	14.705	48.919	*	16.7%

File Size	SCPS-Vegas Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.078	0.086	-0.0080	-0.6910	0.6750		--
10K	7.199	4.231	2.9682	1.8556	4.0808	*	70.2%
100K	27.754	24.250	3.550	-0.328	7.427		--
1000K	193.112	190.625	2.487	-6.490	11.463		--

4.1.2.2 BER=1E-6

Delay of 120 ms

Figure 4.9 shows the protocol performance for channel rate of 115,200 bps:2400 bps, BER=1E-6 and Delay=120 ms. Similar to the plots for asymmetric channel with BER=0 and Delay=120 ms in Figure 4.7, no difference can be seen among the protocols. This tells us that asymmetric channel rate dominates the performance relationship among all protocols and the effects of low error rates and 120 ms delay are not as strong as asymmetric channel rate, even for all file size. The corresponding HSD procedure in Table 4.9 indicates this.

Delay of 1280 ms

The protocol performance comparison relationship for the delay of 1280 ms is shown in Figure 4.10 and Table 4.10. We see the relationship is very similar to that for asymmetric channel with BER=0 and Delay=1280 ms. When the file size gets larger, SCPS-Vegas tends to have better performance than others while the performance means of SCPS-VJ are also significantly different from TCP/IP's. Similar to the case with BER=1E-6, we may see that three protocols show practical performance differences for large files. The performance differences for 10K file and 1000K file between SCPS-Vegas and SCPS-VJ are as we expected earlier.

4.1.2.3 BER=1E-5

Delay of 120 ms

Figure 4.11 displays the protocol performance for asymmetric channel with BER=1E-5 and Delay=120 ms. Basically, we see the lines are almost linear. This is

very similar to the plot for symmetric channel in Figure 4.5 with the difference that, for 1000K file, all the protocols show significant performance difference for asymmetric rate as shown in Table 4.11 and there is significance difference only between TCP/IP and SCPS-VJ for symmetric channel.

Delay of 1280 ms

When the delay is increased to 1280 ms, we see a clear difference between all protocols except for smallest 1K file as shown in Figure 4.12. We also see, as we expected, SCPS-Vegas shows much better performance over SCPS-VJ for 1000K and performs behind SCPS-VJ for 10K file.

The HSD procedure in Table 4.12 indicates that all file comparison pairs except for 1K have significant difference. Comparing to Figure 4.6 where the relationship for symmetric rate is shown, we see when a relatively large file is transmitted with a high error (1E-5) and a large delay (1280 ms), all protocols show practically significant performance difference. This is clearer when the asymmetric slow link is used. For both channel rates, along the file size increases, SCPS-Vegas tends to have best performance and TCP/IP has the slowest throughput.

Conclusions

By summarizing the above comparison results between TCP and SCPS with different delays and BERs under both symmetric and asymmetric channel rates. We conclude that:

- Protocols do not show performance difference with a very small file (≤ 1Kbytes) for all configurations. For both symmetric and asymmetric channel

rates, protocols have no statistically significant performance difference for low BERs with geo-stationary orbit satellite link delay.

- Protocols show statistically and practically significant performance difference with the increase of file size, BER and link delay for both symmetric and asymmetric channel rates and SCPS-Vegas and SCPS-VJ have better performance than TCP/IP does and SCPS-Vegas tends to show the highest throughput. So we reject all **Hypotheses Set 1** to **Hypotheses Set 4** corresponding to the analysis set (1) in Section 3.2.1. We conclude that protocols do not have equal file transfer time means for each of the same transmission conditions.
- But we should note that the conclusion to reject all **Hypotheses Set 1** through **Hypotheses Set 4** does not mean that protocols have significant different file transfer time means for each of the same transmission conditions of link delay, BER and file size with symmetric channel rate. This conclusion actually answers our first two basic questions listed in Chapter 1.
- The answer for question (1) is: There is an overall advantage of the SCPS-Vegas protocol for file transport over TCP/IP in our simulated low BDP satellite channel. The answer for question (2) is: Vegas congestion control mode shows superior performance than VJ based congestion control mechanism based on the performance comparison between SCPS-VJ and SCPS-Vegas.

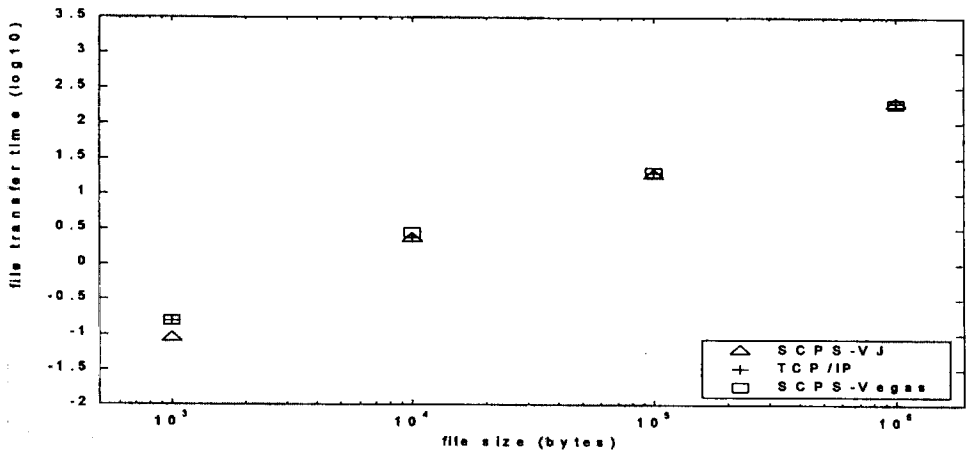

Figure 4.9: File transfer time for TCP/IP, SCPS-VJ and SCSP-Vegas with channel rate of 115,200 bps:2400 bps, BER=1E-6 and Delay=120 ms

Table 4.9: Comparison of means for channel rate of 115,200 bps:2400 bps, BER=1E-6 and Delay=120 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit		Means Difference (%)
1K	0.154	0.088	0.0662	-0.6208	0.7532	--
10K	2.338	2.350	-0.012	-5.844	5.820	--
100K	20.025	19.132	0.893	-6.175	7.961	--
1000K	194.625	192.331	2.294	-14.813	19.402	--
File Size	SCPS-Vegas Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit		Means Difference (%)
1K	0.159	0.088	0.0719	-0.6111	0.7549	--
10K	2.867	2.350	0.5173	-0.5953	1.6299	--
100K	20.027	19.132	0.894	-2.983	4.772	--
1000K	189.916	192.331	-2.414	-11.390	6.562	--

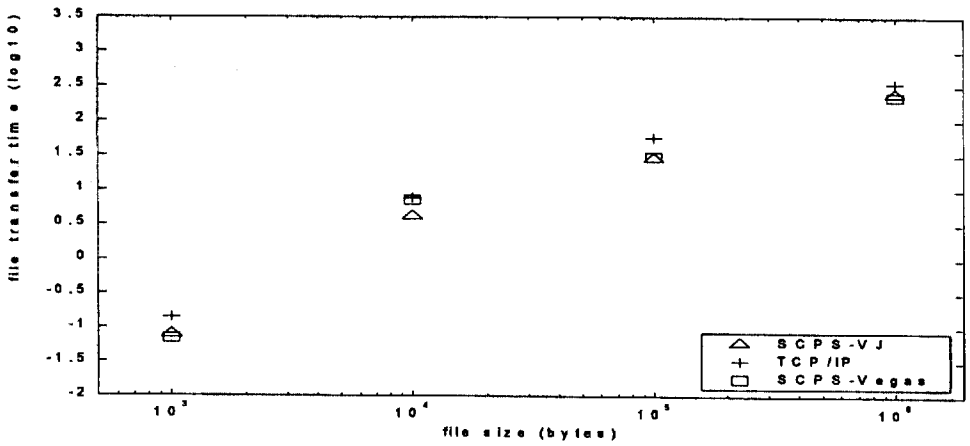

Figure 4.10: File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:2400 bps, BER=1E-6 and Delay=1280 ms

Table 4.10: Comparison of means for channel rate of 115,200 bps:2400 bps, BER=1E-6 and Delay=1280 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.138	0.081	0.0574	-0.6296	0.7444		--
10K	7.657	4.240	3.416	-2.415	9.248		--
100K	57.744	29.338	28.406	21.337	35.474	*	96.8%
1000K	355.813	250.517	105.295	88.188	122.402	*	42%
File Size	SCPS-Vegas Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.068	0.081	-0.0129	-0.6959	0.6701		--
10K	7.245	4.240	3.0042	1.8915	4.1168	*	70.9%
100K	29.958	29.338	0.620	-3.258	4.497		--
1000K	223.997	250.517	-26.520	-35.496	-17.544	*	-10.6%

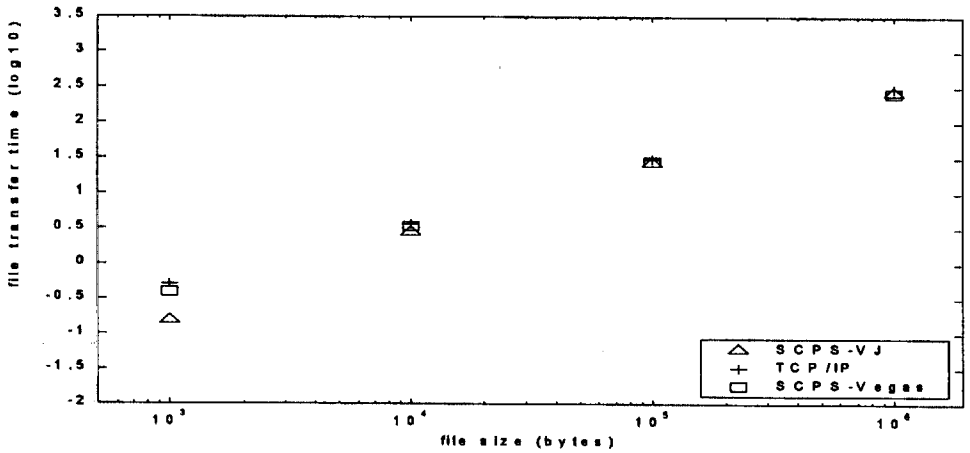

Figure 4.11: File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:2400 bps, BER=1E-5 and Delay=120 ms

Table 4.11: Comparison of means for channel rate of 115,200 bps:2400 bps, BER=1E-5 and Delay=120 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.521	0.158	0.3633	-0.3237	1.0503		--
10K	3.700	2.840	0.860	-4.971	6.692		--
100K	30.925	27.596	3.329	-3.739	10.398		--
1000K	302.563	274.758	27.805	10.698	44.912	*	10.1%
File Size	SCPS-Vegas Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.396	0.158	0.2381	-0.4449	0.9211		--
10K	3.422	2.840	0.5820	-0.5307	1.6946		--
100K	28.595	27.596	0.999	-2.879	4.876		--
1000K	261.204	274.758	-13.554	-22.530	-4.578	*	-4.9%

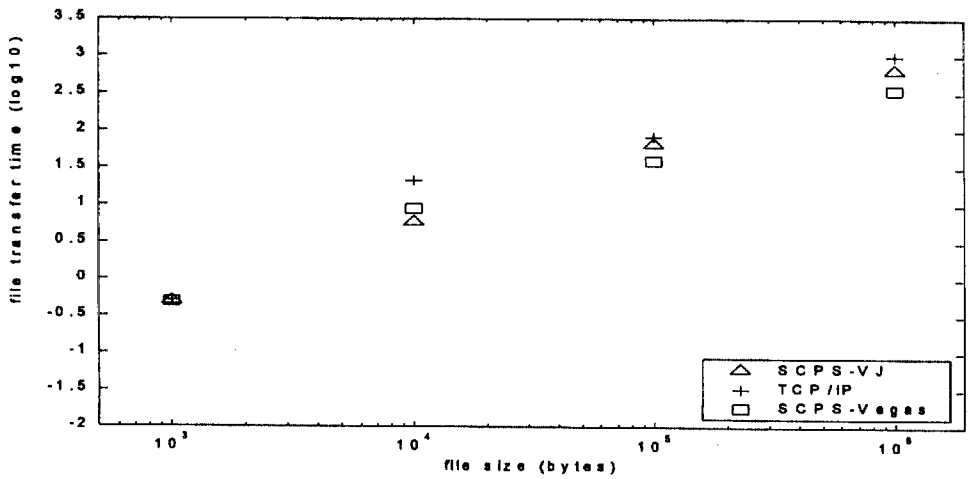

Figure 4.12: File transfer time for TCP/IP, SCPS-VJ and SCPS-Vegas with channel rate of 115,200 bps:2400 bps, BER=1E-5 and Delay=1280 ms

Table 4.12: Comparison of means for channel rate of 115,200 bps:2400 bps, BER=1E-5 and Delay=1280 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.515	0.506	0.0087	-0.6783	0.6957		--
10K	20.935	6.046	14.889	9.057	20.721	*	246.3%
100K	83.744	65.687	18.057	10.988	25.125	*	27.5%
1000K	998.438	669.597	328.840	311.733	345.947	*	49.1%
File Size	**SCPS-Vegas Mean (secs)**	**SCPS-VJ Mean (secs)**	**Means Difference (secs)**	**95% Confidence Limit**			**Means Difference (%)**
1K	0.495	0.506	-0.0112	-0.6942	0.6718		--
10K	8.872	6.046	2.8260	1.7134	3.9386	*	46.7%
100K	40.146	65.687	-25.542	-29.419	-21.664	*	-38.9%
1000K	348.515	669.597	-321.082	-330.058	-312.106	*	-48%

4.2 Delay Effects on Protocol Performance

This section is to investigate how each protocol performs over the link with different delay time and to study how the delays affect the protocol performance. Sections 4.2.1, 4.2.2 and 4.2.3 analyze the delay effects on the performance for each protocol.

4.2.1 Delay Effects on TCP/IP Performance

Symmetric Channel Rate

Figures 4.13-4.15 plot the TCP/IP performance difference with respect to link delay over symmetric channel with each of error rates 0, 1E-6 and 1E-5. By comparing three plots, we see, for all BERs, the file transfer time show no big differences for 1K and 10K files but show statistically significant differences for larger files. Both Mean Differences between two delays are getting larger with the increase of BER from error free to 1E-5. The corresponding HSD procedure in Table 4.13 verifies our observation and conclusion. This indicates that increasing of link delay from 120 ms to 1280 ms significantly affects the TCP/IP performance and the delay effect is becoming stronger along with the increase of BER. The above observation can be understood from the following analyses:

- A small file of 1Kbytes or 10Kbytes which can just be wrapped into less than 10 packets and those limited number of packets are too few to show potential TCP/IP performance. Although with the increase of delay and BER, the performance difference still can not be shown up since the effects of link delay and BER on few packets can not significantly affect the overall performance of TCP/IP.

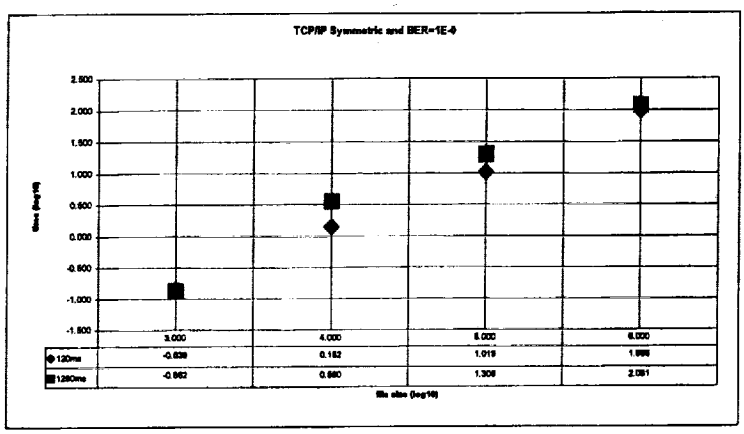

Figure 4.13: TCP/IP performance over symmetric error free channel

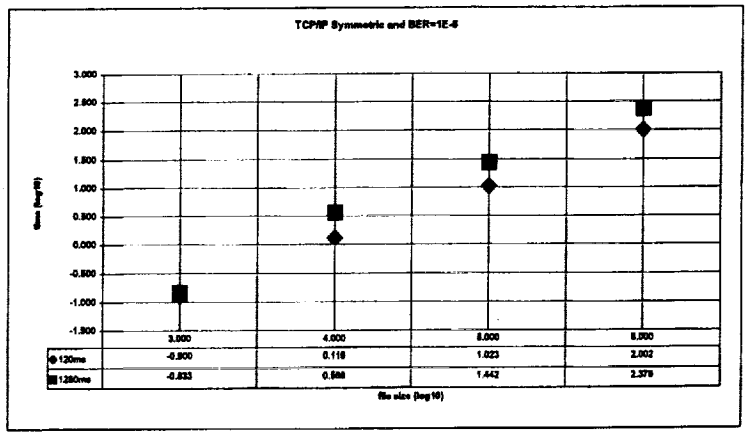

Figure 4.14: TCP/IP performance over symmetric channel with error rate 1E-6

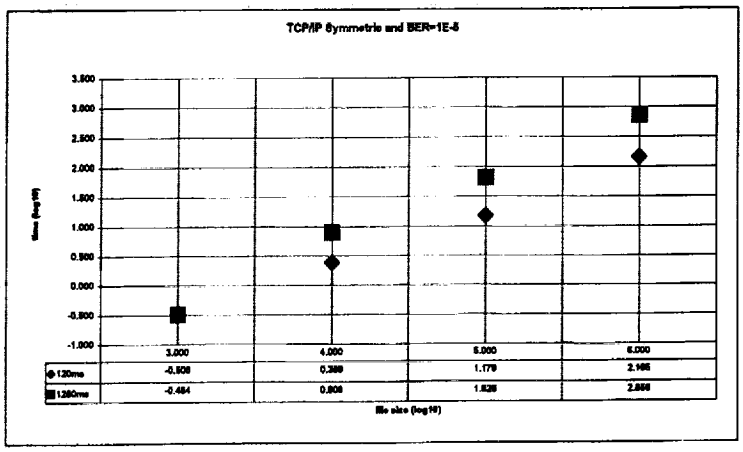

Figure 4.15: TCP/IP performance over symmetric channel with error rate 1E-5

Table 4.13: TCP/IP comparison of means for two delays with channel rate of 115,200 bps:115,200 bps and error rate of 0, 1E-6 and 1E-5

File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	0	0.145	0.137	0.0075	-0.6795	0.6945		--
10K	0	1.420	3.634	-2.214	-8.046	3.618		--
100K	0	10.456	20.244	-9.788	-16.856	-2.719	*	48.4%
1000K	0	96.875	120.438	23.563	-40.670	6.455	*	-19.6%
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-6	0.126	0.147	-0.0210	-0.7080	0.6661		--
10K	1E-6	1.314	3.697	-2.383	-8.215	3.449		--
100K	1E-6	10.544	27.699	-17.124	-24.193	-10.056	*	-61.8%
1000K	1E-6	100.519	239.125	-138.606	-155.713	-121.499	*	-58%
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-5	0.312	0.328	-0.0163	-0.7033	0.6707		--
10K	1E-5	2.450	8.097	-5.647	-11.479	0.185		--
100K	1E-5	15.088	66.919	-51.831	-58.900	-44.763	*	-77.5%
1000K	1E-5	146.063	717.688	-571.625	-588.732	-554.518	*	-79.7%

- But the situation is different for a much large file size (e.g. 100Kbytes or 1000Kbytes) which needs many more packets to complete the file transfer. Along the increase of the packets number, the protocol enters into the steady performance state, the effects of link delay and BER on a large number of packets affect the overall performance and thus TCP/IP performance difference shows up.
- The mean file transfer time difference gets larger with the increase of file size, BER and link delay. This can be understood from the way that a higher BER inserted, more packets corrupted. When more packets are corrupted, two effects are occurred: (1) TCP/IP VJ congestion control algorithm assumes that the congestion occurs and reduces the congestion window and thus slows down the transmission; (2) More retransmissions occur. Effects (1) and (2) definitely cause the protocol to take more file transmission time.
- The above analysis further implies that the facts of the file size, BER, link delay and their interactions contribute more significantly to the variance of the protocol performance than other factors do.

Asymmetric Channel Rate

Figures 4.16-4.18 show delay effects on the TCP/IP performance for asymmetric channel. Similar to the plots for symmetric channel, we cannot see any statistically significant difference for 1K file since it is too small to give the chance

for the protocol to show the performance, and the difference shows up starting from 10K file through all larger files.

By looking at the HSD procedure in Table 4.14 and comparing with symmetric channel plots, we also see the delay effect on the performance gets stronger along with the increase of the file size and the increase of BER. This is identical to the performance over symmetric channel.

The offset problem at 1K point for both symmetric and asymmetric channel rates probably due to the fact that it can just be wrapped into one packet and runs with a single window and not full interaction.

Combining the above analyses for both symmetric and asymmetric channels, we may have the following conclusions for the delay effects on TCP/IP:

- For both symmetric and asymmetric channels with all BERs, TCP/IP shows no statistically significant difference of the transfer time with the change of link delay for a very small file (1K).
- Significant performance difference shows up along the increase of the file size and becomes larger when the error rate is increased.
- The performance difference due to the increase of the file size and BER is stronger for asymmetric channel than symmetric channel.

4.2.2 Delay Effects on SCPS-VJ Performance

Symmetric Channel Rate

Figures 4.19-4.21 display the delay effects on the SCP-VJ performance for symmetric channel with all BERs.

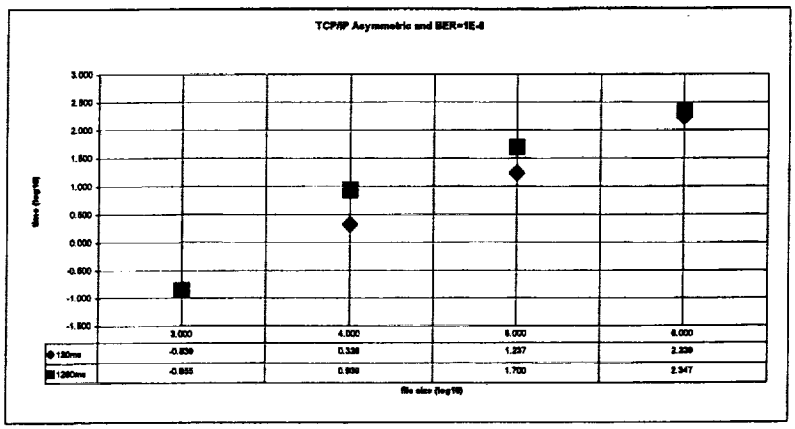

Figure 4.16: TCP/IP performance over asymmetric error free channel

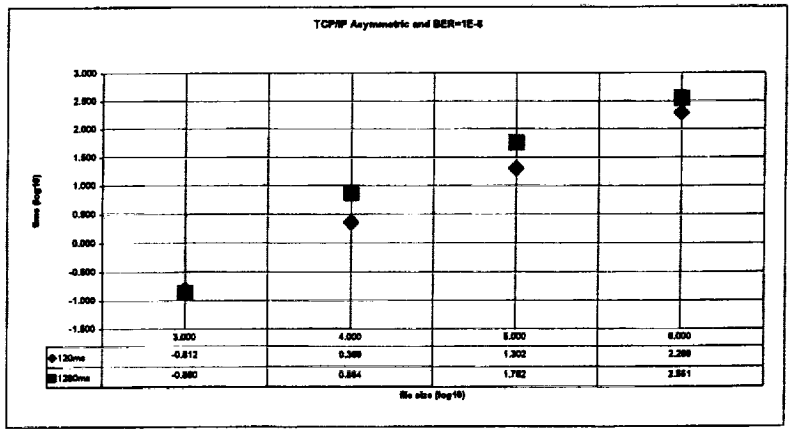

Figure 4.17: TCP/IP performance over asymmetric channel with error rate 1E-6

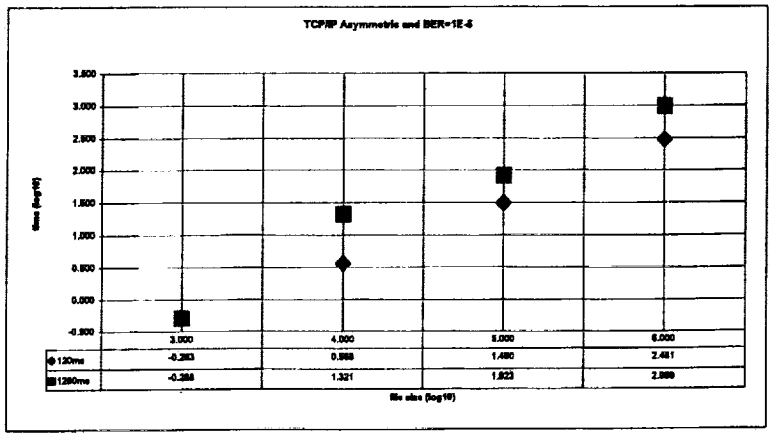

Figure 4.18: TCP/IP performance over asymmetric channel with error rate 1E-5

Table 4.14: TCP/IP comparison of means for two delays with channel rate of 115,200 bps:2400 bps and error rate of 0, 1E-6 and 1E-5

File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	0	0.145	0.140	0.0058	-0.6812	0.6928		--
10K	0	2.128	8.689	-6.561	-12.393	-0.729	*	-75.5%
100K	0	17.269	50.156	-32.888	-39.956	-25.819	*	-65.6%
1000K	0	173.500	222.438	-48.938	-66.045	-31.830	*	-22%
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-6	0.154	0.138	0.0158	-0.6713	0.7028		--
10K	1E-6	2.338	7.657	-5.319	-11.151	0.513		--
100K	1E-6	20.025	57.744	-37.719	-44.787	-30.650	*	-65.3%
1000K	1E-6	194.625	355.813	-161.188	-178.295	-144.080	*	-45.3%
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-5	0.521	0.515	0.0066	-0.6805	0.6936		--
10K	1E-5	3.700	20.935	-17.235	-23.067	-11.403	*	-82.3%
100K	1E-5	30.925	83.744	-52.819	-59.887	-45.750	*	-63.1%
1000K	1E-5	302.563	998.438	-695.875	-712.982	-678.768	*	-69.7%

Similar to TCP/IP, SCPS-VJ shows no statistically significant performance difference with respect to the change of link delay for small files and significant difference is shown up when the file size gets large, and the difference becomes large along the increase of BER. For BER=1E-5, two performance lines seem to be parallel each other for all file size. The HSD procedure in Table 4.15 indicates that there is about one order of magnitude time difference between two delays along the file size.

Asymmetric Channel Rate

The asymmetric channel plots for each of three BERs are shown in Figures 4.22-4.24. Similar to all previous plots, the protocol shows no statistically significant means difference with the change of link delay for small file size and significant difference is occurred for large files and is getting large along the increase of the file size and of the BER. Identical to symmetric channel, with BER=1E-5, two performance lines tend to be parallel with one order of magnitude spaced along the file size increase. The HSD procedure in Table 4.16 provides the comparison of means corresponding to Figures 4.22-4.24.

In summary, the conclusions (1) and (2) obtained in Section 4.2.1 for TCP/IP are also valid for SCPS-VJ. Different from TCP/IP which shows more strong performance difference for asymmetric channel than symmetric channel, SCPS-VJ's performance difference between two delays tends to get smaller for asymmetric channel, i.e., SCPS-VJ is better behaved on asymmetric channels. These can be clearly seen when we compare Table 4.15 and Table 4.16.

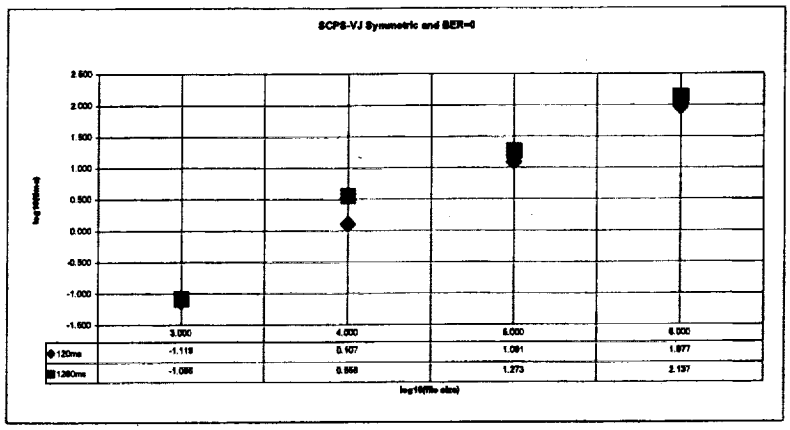

Figure 4.19: SCPS-VJ performance over symmetric error free channel

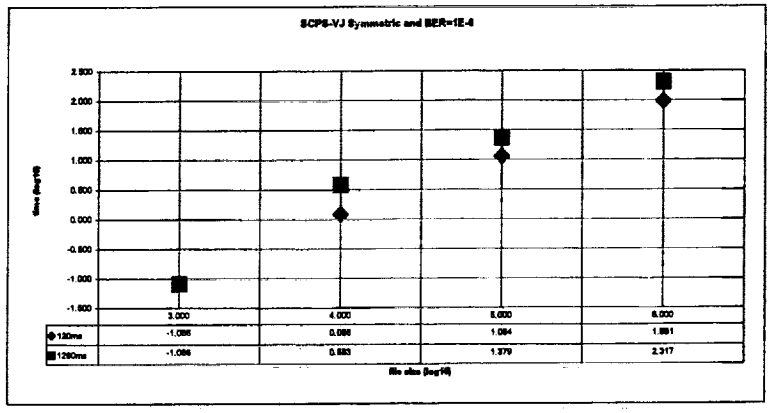

Figure 4.20: SCPS-VJ performance over symmetric channel with error rate 1E-6

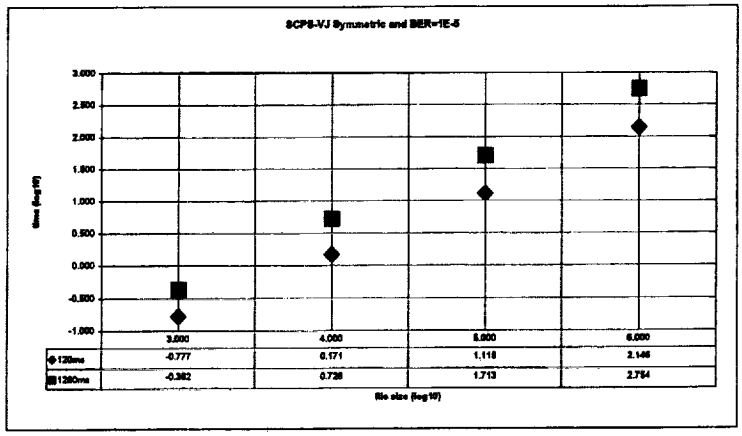

Figure 4.21: SCPS-VJ performance over symmetric channel with error rate 1E-5

Table 4.15: SCPS-VJ comparison of means for two delays with channel rate of 115,200 bps:115,200 bps and error rate of 0, 1E-6 and 1E-5

File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	0	0.076	0.082	-0.0060	-0.6930	0.6810		--
10K	0	1.278	3.612	-2.334	-8.166	3.498		--
100K	0	12.321	18.729	-6.407	-13.476	0.661		--
1000K	0	94.823	137.043	-42.220	-59.327	-25.113	*	-30.8%
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-6	0.082	0.082	-0.0005	-0.6876	0.6865		--
10K	1E-6	1.219	3.829	-2.609	-8.441	3.223		--
100K	1E-6	11.585	23.929	-12.344	-19.412	-5.275	*	-51.6%
1000K	1E-6	97.910	207.292	-109.382	-126.489	-92.275	*	-52.7%
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-5	0.167	0.435	-0.2679	-0.9549	0.4192		--
10K	1E-5	1.484	5.340	-3.857	-9.688	1.975		--
100K	1E-5	13.023	51.649	-38.626	-45.694	-31.557	*	-74.8%
1000K	1E-5	139.901	567.325	-427.424	-444.531	-410.317	*	-75.3%

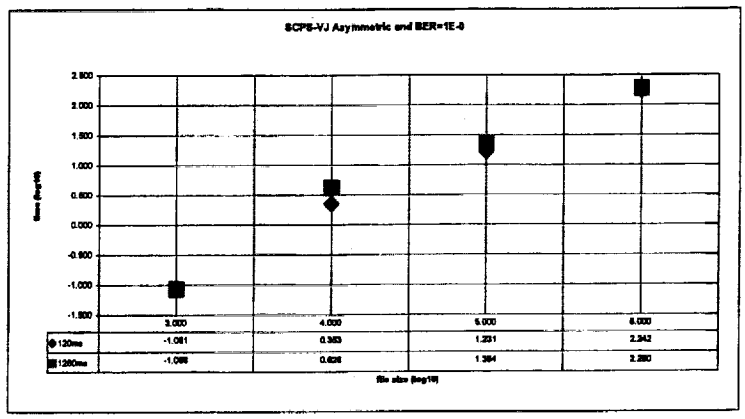

Figure 4.22: SCPS-VJ performance over asymmetric error free channel

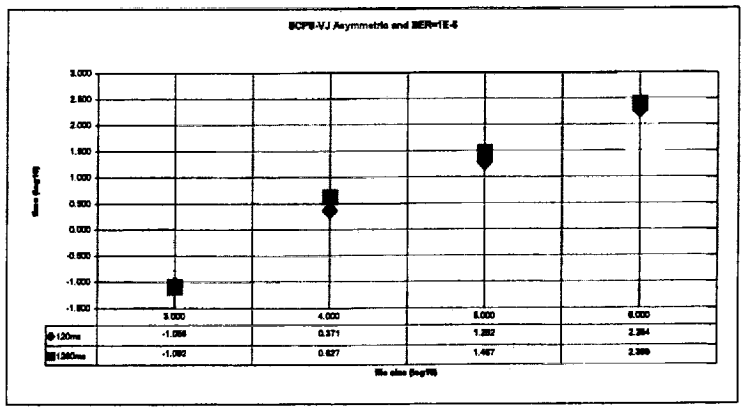

Figure 4.23: SCPS-VJ performance over asymmetric channel with error rate 1E-6

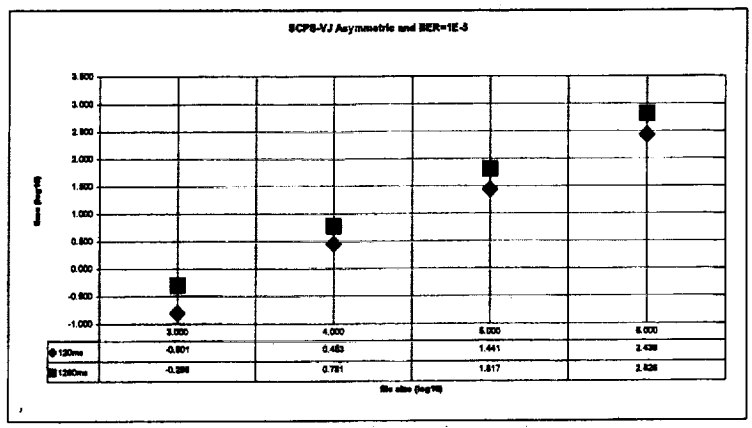

Figure 4.24: SCPS-VJ performance over asymmetric channel with error rate 1E-5

Table 4.16: SCPS-VJ comparison of means for two delays with channel rate of 115,200 bps:2400 bps and error rate of 0, 1E-6 and 1E-5

File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	0	0.089	0.086	0.0032	-0.6839	0.6902		--
10K	0	2.255	4.231	-1.977	-7.808	3.855		--
100K	0	17.022	24.205	-7.182	-14.251	-0.114	*	-29.7%
1000K	0	174.482	190.625	-16.143	-33.250	0.964		
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-6	0.088	0.081	0.0069	-0.6801	0.6940		--
10K	1E-6	2.350	4.240	-1.890	-7.722	3.942		--
100K	1E-6	19.132	29.338	-10.206	-17.274	-3.138	*	-34.8%
1000K	1E-6	192.331	250.517	-58.187	-75.294	-41.079	*	-23.2%
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-5	0.158	0.506	-0.3480	-1.0351	0.3390		--
10K	1E-5	2.840	6.046	-3.206	-9.038	2.626		--
100K	1E-5	27.596	65.687	-38.091	-45.160	-31.023	*	-58%
1000K	1E-5	274.758	669.597	-394.840	-411.947	-377.732	*	-59%

4.2.3 Delay Effects on SCPS-Vegas Performance

Symmetric Channel Rate

Figures 4.25-4.27 plot the SCPS-Vegas performance for two delays with three BERs. Similar to Figures 4.13-4.15 for TCP/IP and Figures 4.19-4.21 for SCPS-VJ, SCPS-Vegas have no significant difference with the change of delay for 1K file. The difference is that SCPS-Vegas shows difference with the change of delay starting from 10K up to all larger files where neither TCP/IP or SCPS-VJ show difference for 10K. Along the increase of the file size, the significant difference becomes large when the error rate is high. Table 4.17 indicates that SCPS-Vegas's performance is significantly different between the two delays for all 10K, 100K and 1000K files.

Asymmetric Channel Rate

Figures 4.28-4.30 show three plots of SCPS-Vegas performance over asymmetric channel with two delays. By looking at Table 4.18, we see, similar to the plots in Figures 4.25-4.27, all three plots have no significant difference with respect to delay change for 1K file but shows significant difference for all files starting from 10K up to all larger files. An exception occurs for 1K file with asymmetric channel and BER=1E-6 in Figure 4.29, which shows the transfer with delay of 120ms takes more time than the one with 1280 ms. This is caused by one run with delay of 120 ms which takes exceptionally long time while the average of other fifteen runs is very close to the average of the total sixteen runs with the delay of 1280 ms. This can also be verified by that the standard deviation of the transfer time with delay of 120 ms is about 29 times large of the one with 1280 ms delay.

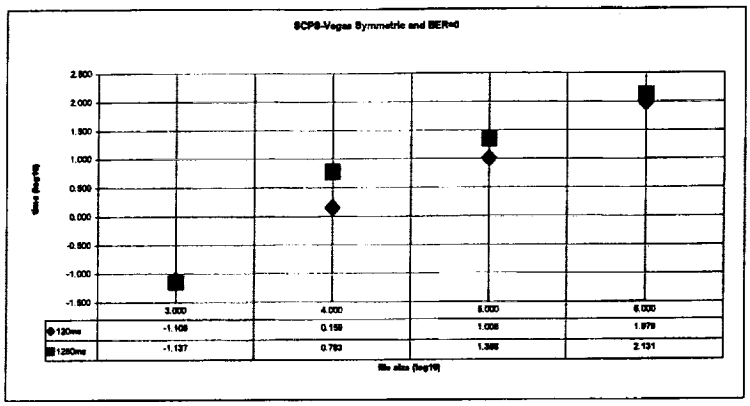

Figure 4.25: SCPS-Vegas performance over symmetric error free channel

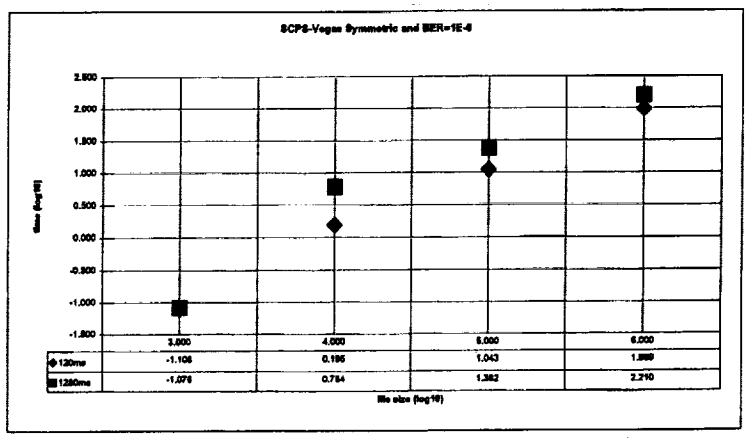

Figure 4.26: SCPS-Vegas performance over symmetric channel with error rate 1E-6

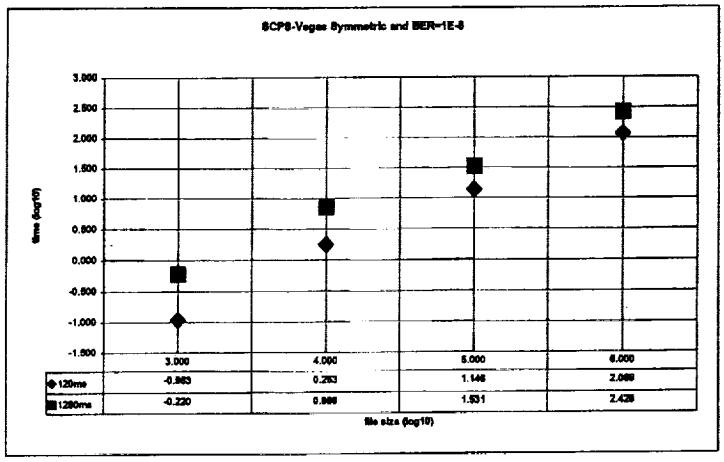

Figure 4.27: SCPS-Vegas performance over symmetric channel with error rate 1E-5

Table 4.17: SCPS-Vegas comparison of means for two delays with channel rate of 115,200 bps:115,200 bps and error rate of 0, 1E-6 and 1E-5

File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	0	0.078	0.073	0.0048	-0.6782	0.6878		--
10K	0	1.442	6.074	-4.6319	-5.7445	-3.5192	*	-76.3%
100K	0	10.190	22.675	-12.485	-16.362	-8.608	*	-55%
1000K	0	95.351	135.183	-39.832	-48.808	-30.856	*	-29.5%

File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-6	0.078	0.084	-0.0057	-0.6887	0.6773		--
10K	1E-6	1.568	6.086	-4.5181	-5.6307	-3.4054	*	-74.2%
100K	1E-6	11.052	24.102	-13.050	-16.927	-9.173	*	-54.1%
1000K	1E-6	97.402	162.009	-64.607	-73.583	-55.631	*	-39.9%

File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-5	0.109	0.603	-0.4940	-1.1770	0.1890		--
10K	1E-5	1.789	7.351	-5.5614	-6.6741	-4.4488	*	-75.7%
100K	1E-5	14.006	33.944	-19.938	-23.815	-16.060	*	-58.7%
1000K	1E-5	117.240	267.870	-150.630	-159.606	-141.653	*	-56.2%

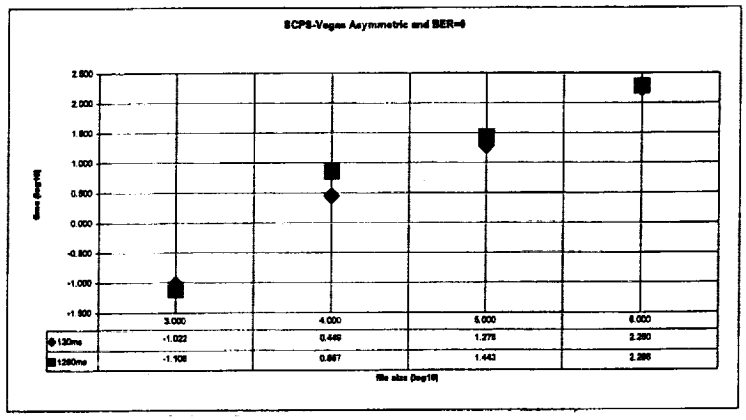

Figure 4.28: SCPS-Vegas performance over asymmetric error free channel

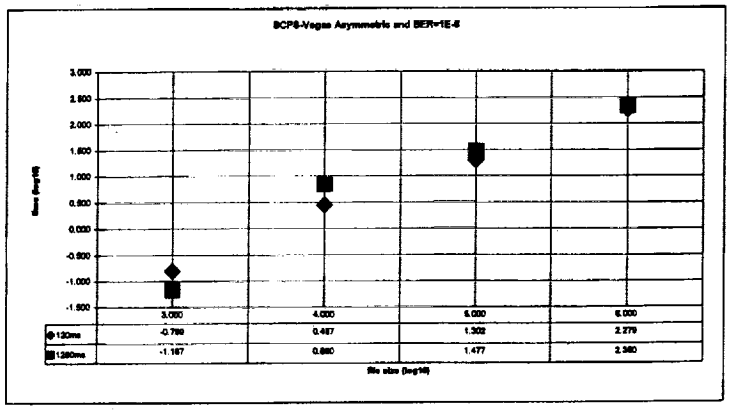

Figure 4.29: SCPS-Vegas performance over asymmetric channel with error rate 1E-6

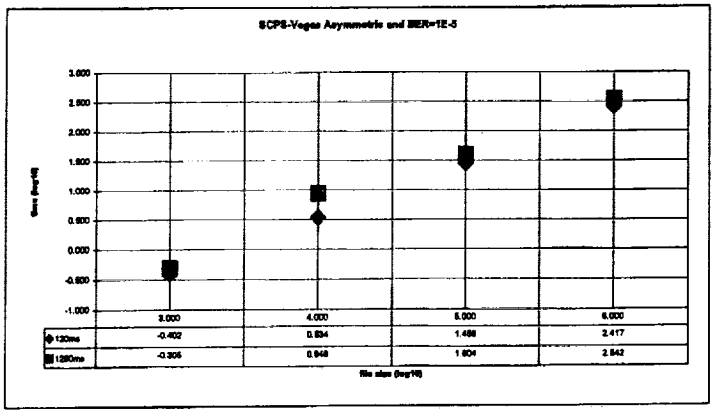

Figure 4.30: SCPS-Vegas performance over asymmetric channel with error rate 1E-5

Table 4.18: SCPS-Vegas comparison of means for two delays with channel rate of 115,200 bps:2400 bps and error rate of 0, 1E-6 and 1E-5

File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	0	0.095	0.078	0.0176	-0.6654	0.7006		--
10K	0	2.814	7.199	-4.3857	-5.4983	-3.2731	*	-61%
100K	0	18.976	27.754	-8.778	-12.655	-4.901	*	-31.6%
1000K	0	177.758	193.112	-15.354	-24.330	-6.377	*	-8%
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-6	0.159	0.068	0.0917	-0.5913	0.7748		--
10K	1E-6	2.867	7.245	-4.3772	-5.4898	-3.2645	*	-60.4%
100K	1E-6	20.027	29.958	-9.931	-13.809	-6.054	*	-33.1%
1000K	1E-6	189.916	223.997	-34.081	-43.057	-25.105	*	-15.2%
File Size	BER	120ms Mean (secs)	1280ms Mean (secs)	Mean Difference (secs)	95% Confidence Limit			Mean Difference (%)
1K	1E-5	0.396	0.495	-0.0988	-0.7818	0.5843		--
10K	1E-5	3.422	8.872	-5.4502	-6.5628	-4.3375	*	-61.4%
100K	1E-5	28.595	40.146	-11.551	-15.428	-7.674	*	-28.8%
1000K	1E-5	261.204	348.515	-87.312	-96.288	-78.335	*	-25.1%

For SCPS-Vegas, we may have the following conclusions for the delay effects on the performance:

- SCPS-Vegas has no significant performance differences for 1K file but shows significant performance difference for file ranging from 10K to 1000K. The difference gets large along the increase of file size and BER.
- By comparing SCPS-Vegas with TCP/IP and SCPS-VJ, we see SCPS-Vegas's performance is more sensitive to the increase of BER, even for the symmetric channel and smaller file size (10K). Both TCP/IP and SCPS-VJ show no significant performance difference for 10K file when symmetric channel rate is used.
- Similar to SCPS-VJ, SCPS-Vegas shows a larger performance difference between two delays for symmetric channel rate than for asymmetric channel. This is different from TCP/IP which shows a larger difference when asymmetric channel is used.
- This may be considered as one of key differences between SCPS implementations and TCP/IP.

Conclusions

Based on the above study of the link delay effects on the performance of each protocol, we conclude that:

- Similar to the result for protocol performance comparison, for both symmetric and asymmetric channel rates, all protocols do not show statistically

significant performance difference with respect the link delay change for a very small file (≤ 1Kbytes) with all three BERs.

- All protocols show statistically significant different performance with respect to the link delay change along the increase of file size. The difference becomes practical and more significant when the error rate is increased. Based on this result, we reject all **Hypotheses Set 1** through **Hypotheses Set 3** corresponding to the analysis set (2) in Section 3.2.1 and conclude that all protocols do not have equal file transfer time means with Delay=120 ms and Delay=1280 ms for each of the same transmission conditions of channel rate, BER and file size. Similar to rejecting hypotheses when we compare the protocol performance between protocols in Section 4.1, this does not mean that protocols have significant different file transfer time means with Delay=120 ms and Delay=1280 ms for each of the same transmission conditions of channel rate, BER and file size. We know that 1K file shows no significant performance differences with respect to delay change for all configuration.
- TCP performance difference due to the link delay change along with the increase of the file size and BER is larger for asymmetric channel than for symmetric channel; SCPS-VJ and SCPS-Vegas show this difference being stronger for symmetric channel rate. This may be considered as one of key differences between TCP and SCPS implementations.

- The above conclusion partially addresses our basic question (3) on how link delay affects each protocol performance.

4.3 Bit-Error-Rate (BER) Effects on Protocol Performance

In this section, the BER effect on each protocol performance over SGLS test-bed is studied. It is expected to investigate how each protocol performs differently over the simulated channel with the BERs of 0, 1E-6 and 1E-5. Sections 4.3.1, 4.3.2 and 4.3.2 analyze the BER effect on the protocol performance for each of TCP/IP, SCPS-VJ and SCPS-Vegas. For each protocol, the analysis is done for each of the joint transmission condition with each of symmetric/asymmetric channel rates and each of two link delays.

The linear regression models of the file transfer time for each protocol using BER, link delay and file size as regressors are also built over each channel rate. In the course of our analysis, we observed that logarithmic file transfer time was better behaved than just file transfer time. This led us to fit linear regression models to $\log_{10}(Time)$, using the quantified parameters as regressors, and fitting different regressions to each combination of protocol and channel rate. We fit all models simultaneously to make coherent conclusion about global coverage of all protocols and channel rates for the experiments over the SGLS test-bed.

We have a global R^2=0.9503. We know R^2 measures how much variation in the dependent variables can be accounted for by the model. With value ranging from 0 to 1, in general, the larger the R^2 value, the better the model fits the data. For our

models, we have a very high value for R^2, this tells us that our models fit the experiment data very well.

We used the SAS General Linear Model (GLM) and REG procedures to fit and assess the models.

The GLM procedure is the most general analysis-of-variance procedure, which can be used for many different analyses including analysis of variance, regressions and analysis of covariance. It uses the method of least squares to fit general linear models.

The REG procedure is a general-purpose procedure for regression. The REG procedure also uses the principle of least squares to produce estimates that are the best linear unbiased estimates under classical statistical assumptions. Both GLM and REG procedures include a number of useful diagnostic tools for assessing regression models [28], [29], [30].

After fitting a full linear regression models, we also checked to see if we could pool across one or more protocols and/or channel types. When we restricted the model, we noticed that the model surface did not pass near the data points although R^2 was not decreased much. We concluded that pooling over protocol and channel type was not a good idea.

4.3.1 Bit-Error-Rate Effects on TCP/IP

4.3.1.1 Symmetric Channel Rate

Based on the experimental data, the TCP/IP linear regression model using BER, link delay and file size as regressors is built for symmetric channel rate.

$\log_{10}(Time) = -3.576 - 54562.14 \times BER + 0.915 \times \log_{10}(FS) - 0.00014 \times Delay$

$+0.00008 \times Delay \times \log_{10}(FS) + 25.385 \times BER \times Delay + 13633.999 \times BER \times \log_{10}(FS)$.

The values and unit of each parameter contained in our models are listed below:

Time — Averaged file transfer time (second)

FS — File size (bytes) ranging from 1000, 10,000, 100,000 to1,000,000

Delay — Link delay time (millisecond), which is either 120 or 1280 in our experiments

BER — Bit-Error-Rate ranging from 0, 1E-6 to 1E-5.

The above values and unit of each parameter are valid for all model built when we study the experiments over SGLS test-bed.

Delay of 120 ms

Figure 4.31 plots the TCP/IP performance over a symmetric channel with different BERs. We see BER=0 and BER=1E-6 track together while BER=1E-5 rides above them and keeps almost parallel. We also see that three straight least-squares trend lines representing three different BERs fit the data very well.

Table 4.19 indicates that TCP/IP has significant performance difference between BER=1E-6 and BER=1E-5 at the point of 1000K file. This verifies our previous observation that the protocol shows significantly different performance with the increase of file size and the increase of BER. This is reasonable and is to be expected.

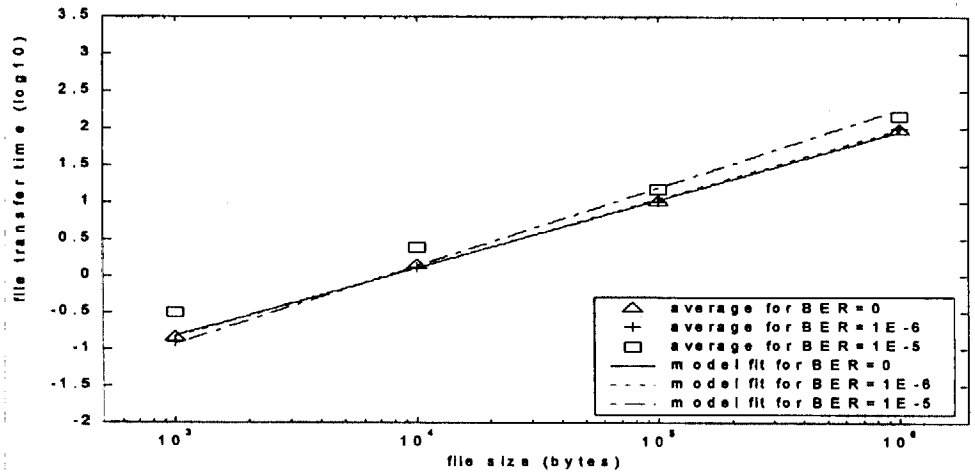

Figure 4.31: TCP/IP performance for different BERs over symmetric channel with a delay of 120 ms

Table 4.19: TCP/IP comparison of means for different BERs over symmetric channel with a delay of 120 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.145	0.126	0.0186	-0.6684	0.7057		--
10K	1.420	1.314	0.106	-5.726	5.938		--
100K	10.456	10.544	-0.088	-7.157	6.980		--
1000K	96.875	100.519	-3.644	-20.751	13.463		--
File Size	BER=1E-6 Mean (secs)	BER=1E-5 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.126	0.312	-0.1857	-0.8727	0.5013		--
10K	1.314	2.450	-1.136	-6.968	4.696		--
100K	10.544	15.088	-4.543	-11.612	2.525		--
1000K	100.519	146.063	-45.544	-62.651	-28.437	*	-31.2%

Delay of 1280 ms

When the link delay is increased to 1280 ms, three TCP/IP performance averages and regression lines tend to split for larger files and keep similar to the delay of 120 ms for small files as shown in Figure 4.32. Table 4.20 indicates that TCP/IP has significant difference among BERs for 100K and 1000K. Based on the above analyses, we have the following conclusions for TCP/IP over symmetric channel:

- The increase of BER from error free to 1E-6 does not significantly affect the TCP/IP's performance when the file size is small. TCP/IP performs significant differently with a BER of 1E-5 is used to transmit a large file.
- For delay of 1280 ms, TCP/IP shows different performance with different BERs for large files. The joint conditions of a large file, a long delay makes protocols show significant decrease of throughput with the change of BER.

4.3.1.2 Asymmetric Channel Rate

The TCP/IP linear regression model for asymmetric channel rate is built below using BER, link delay and file size as regressors:

$$\log_{10}(Time) = -3.76 - 12948.667 \times BER + 0.995 \times \log_{10}(FS) - 0.00008 \times Delay$$
$$+ 0.00007 \times Delay \times \log_{10}(FS) + 8.27 \times BER \times Delay + 7391.65 \times BER \times \log_{10}(FS)$$

When we compare TCP/IP models for both channel rates, we see that BER, link delay and file size contribute more significantly to file transfer time in asymmetric model. This verifies our conclusion when we study the delay effects that TCP performance difference due to the link delay change along the increase of the file size and BER is larger for asymmetric channel than for symmetric channel.

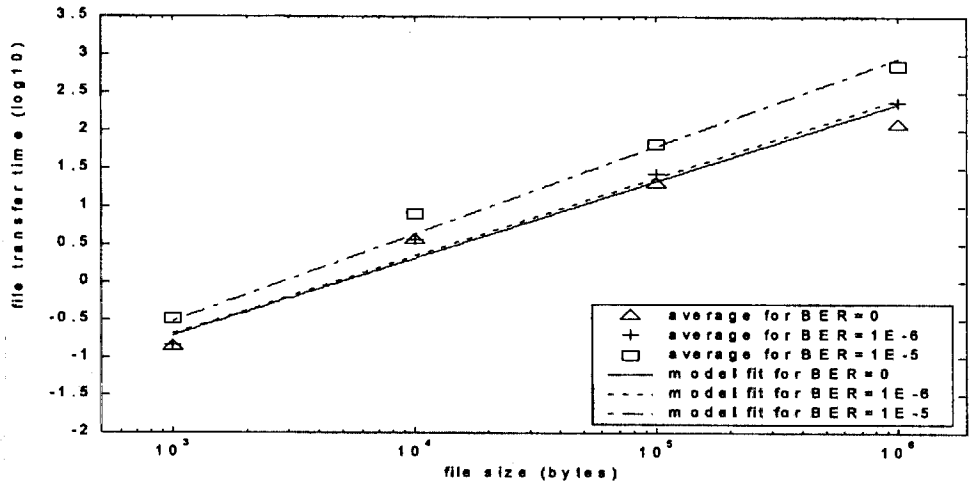

Figure 4.32: TCP/IP performance for different BERs over symmetric channel with a delay of 1280 ms

Table 4.20: TCP/IP comparison of means for different BERs over symmetric channel with a delay of 1280 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.137	0.147	-0.0098	-0.6968	0.6772		--
10K	3.634	3.697	-0.063	-5.895	5.769		--
100K	20.244	27.669	-7.425	-14.493	-0.357	*	-26.8%
1000K	120.438	239.125	-118.688	-135.795	-101.580	*	-49.6%
File Size	BER=1E-6 Mean (secs)	BER=1E-5 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.147	0.328	-0.1810	-0.8680	0.5060		--
10K	3.697	8.097	-4.400	-10.232	1.432		--
100K	27.669	66.919	-39.250	-46.318	-32.182	*	-58.7%
1000K	239.125	717.688	-478.563	-495.670	-461.455	*	-66.7%

Delay of 120 ms

When a delay of 120 ms used over asymmetric channel, similar to Figure 4.31, BER=1E-5 is far above BER=0 and BER=1E-6 with the difference that they are spaced wider each other along the increase of the file size as shown in Figure 4.33. We see the model fit the averages well. Table 4.21 displays that TCP/IP performs differently among BERs for 1000K and between error free and others for 100K.

Delay of 1280 ms

For asymmetric channel rate with a delay of 1280 ms, similar to a delay of 1280 ms for symmetric channel in Figure 4.32, BER=0 and BER=1E-6 locate closely while BER=1E-5 is away from them shown in Figure 4.34. The straight observation shows that three BERs tend to have different performance for large files. Table 4.22 verifies the observation. We also see the models do not fit well at 10K points especially with BER=1E-5. In summary, we have:

- BER=1E-5 tends to reduce the link throughput much more strongly than BER=0 and BER=1E-6.
- With the link delay of 120 ms, increasing BER from 1E-6 to 1E-5 shows practical different performance when transmitting 1000K; with link delay of 1280 ms, practical different performance show up for increasing BER from 0 to 1E-6 when transmitting 1000K file and for increasing BER from 1E-6 to 1E-5 when transmitting 100K and 1000K files. This verifies our previous conclusion that the increases of file size, long delay and BER makes protocol show significantly different performance.

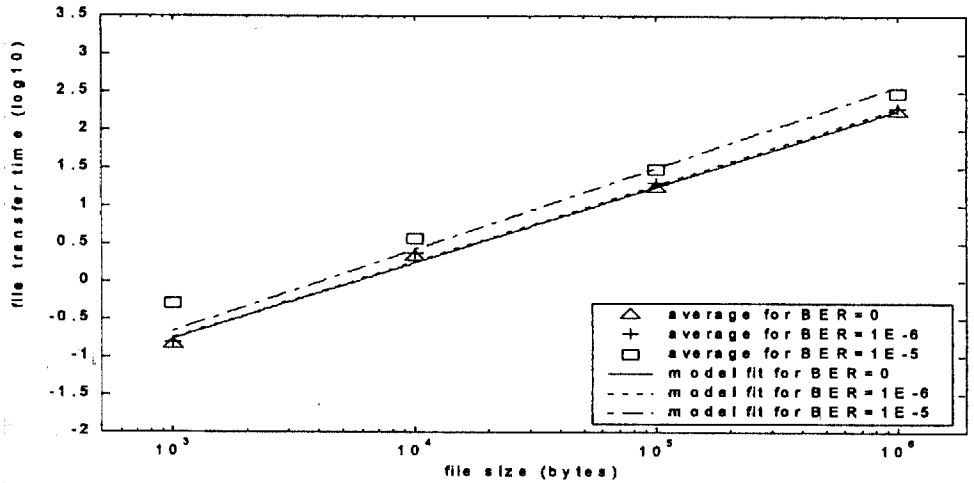

Figure 4.33: TCP/IP performance for different BERs over asymmetric channel with a delay of 120 ms

Table 4.21: TCP/IP comparison of means for different BERs over asymmetric channel with a delay of 120 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.145	0.154	-0.0085	-0.6955	0.6785		--
10K	2.128	2.338	-0.210	-6.042	5.622		--
100K	17.269	20.025	-2.756	-9.825	4.312		--
1000K	173.500	194.625	-21.125	-38.232	-4.018	*	-10.9%
File Size	**BER=1E-6 Mean (secs)**	**BER=1E-5 Mean (secs)**	**Means Difference (secs)**	**95% Confidence Limit**			**Means Difference (%)**
1K	0.154	0.521	-0.3673	-1.0543	0.3198		--
10K	2.338	3.700	-1.362	-7.194	4.470		--
100K	20.025	30.925	-10.900	-17.968	-3.832	*	-35.2%
1000K	194.625	302.563	-107.938	-125.045	-90.830	*	-35.7%

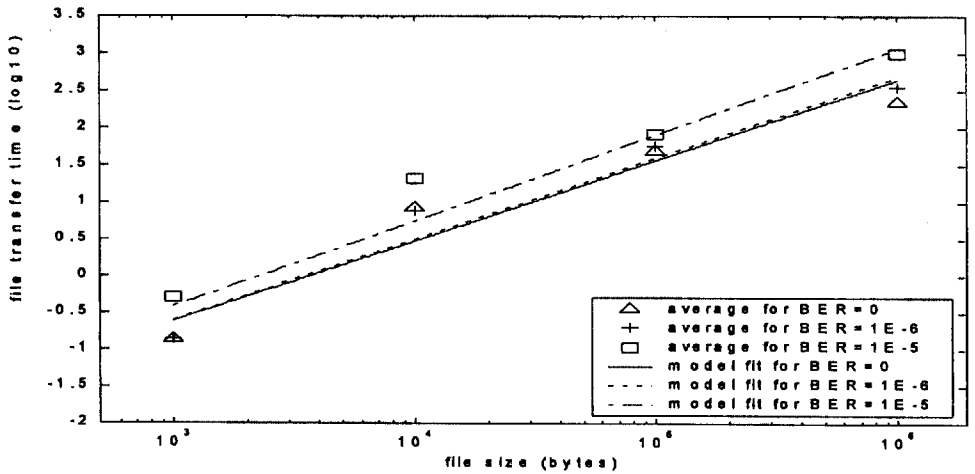

Figure 4.34: TCP/IP performance for different BERs over asymmetric channel with a delay of 1280 ms

Table 4.22: TCP/IP comparison of means for different BERs over asymmetric channel with a delay of 1280 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.140	0.138	0.0014	-0.6856	0.6885		--
10K	8.689	7.657	1.033	-4.799	6.864		--
100K	50.156	57.744	-7.588	-14.656	-0.519	*	-13.1%
1000K	222.438	355.813	-133.375	-150.482	-116.268	*	-37.5%
File Size	BER=1E-6 Mean (secs)	BER=1E-5 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.138	0.515	-0.3764	-1.0635	0.3106		--
10K	7.657	20.935	-13.278	-19.110	-7.446	*	-63.4%
100K	57.744	83.744	-26.000	-33.068	-18.932	*	-31%
1000K	355.813	998.438	-642.625	-659.732	-625.518	*	-64.4%

4.3.2 Bit-Error-Rate Effects on SCPS-VJ

4.3.2.1 Symmetric Channel Rate

The SCPS-VJ linear regression model with symmetric channel rate using BER, link delay and file size as regressors is built below:

$$\log_{10}(Time) = -4.08 - 13794.46 \times BER + 1.01 \times \log_{10}(FS) - 0.00015 \times Delay$$
$$+0.00008 \times Delay \times \log_{10}(FS) + 17.32 \times BER \times Delay + 5265.39 \times BER \times \log_{10}(FS)$$

Delay of 120 ms

Figure 4.35 plots the BER effect on SCPS-VJ performance for symmetric channel and a delay of 120 ms. Similar to the corresponding plot for TCP/IP in Figure 4.31, all averages and three linear regression lines tend to bind together. The only significant performance difference is shown up between BER=1E-6 and BER=1E-5 at the point of 1000K file, which is identical to TCP/IP. This is indicated by the HSD procedure in Table 4.23. This is very reasonable and to be expected.

Delay of 1280 ms

Figure 4.36 shows how the performance is affected when the delay is increased to 1280 ms. Similar to TCP/IP in Figure 4.32, the protocol performance points tend to leave each other and show the performance difference along the increase of file size. Regression lines with BER=0 and BER=1E-6 tight closely while the line with BER=1E-5 is away them.

Different from Figure 4.35, the performance difference among three BERs are much smaller when we compare Table 4.23 and Table 4.24.

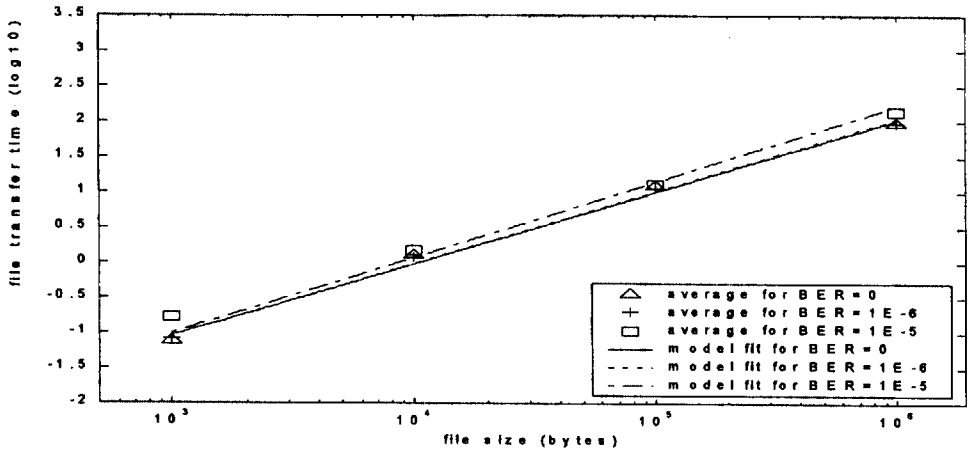

Figure 4.35: SCPS-VJ performance for different BERs over symmetric channel with a delay of 120 ms

Table 4.23: SCPS-VJ comparison of means for different BERs over symmetric channel with a delay of 120 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit		Means Difference (%)
1K	0.076	0.082	-0.0055	-0.6926	0.6815	--
10K	1.278	1.219	0.059	-5.773	5.8911	--
100K	12.321	11.585	0.736	-6.332	7.805	--
1000K	94.823	97.910	-3.088	-20.195	14.020	--
File Size	**BER=1E-6 Mean (secs)**	**BER=1E-5 Mean (secs)**	**Means Difference (secs)**	**95% Confidence Limit**		**Means Difference (%)**
1K	0.082	0.167	-0.0855	-0.7725	0.6015	--
10K	1.219	1.484	-0.264	-6.096	5.568	--
100K	11.585	13.023	-1.438	-8.506	5.630	--
1000K	97.910	139.901	-41.990	-59.097	-24.883 *	-30%

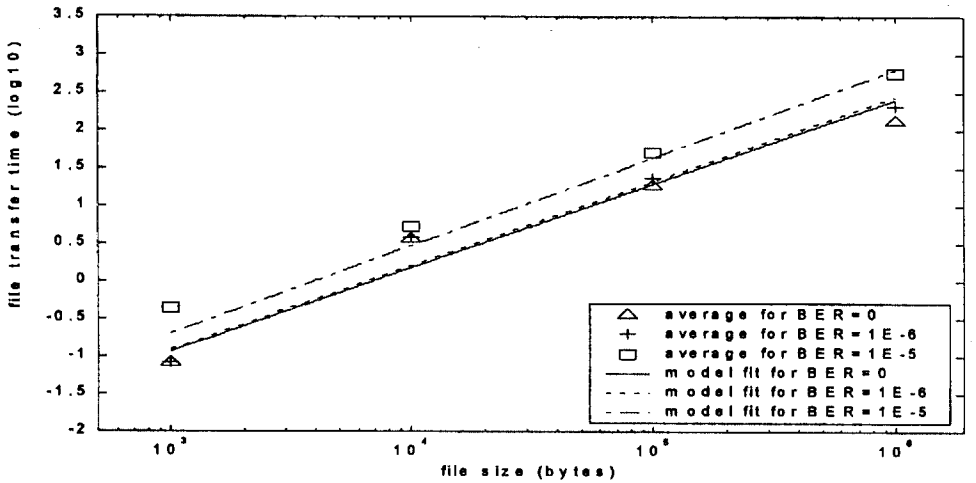

Figure 4.36: SCPS-VJ performance for different BERs over symmetric channel with a delay of 1280 ms

Table 4.24: SCPS-VJ comparison of means for different BERs over symmetric channel with a delay of 1280 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.082	0.082	-0.0001	-0.6871	0.6869		--
10K	3.612	3.829	-0.216	-6.048	5.616		--
100K	18.729	23.929	-5.200	-12.269	1.868		--
1000K	137.034	–207.292	70.250	-87.357	-53.142	*	-33.9%
File Size	**BER=1E-6 Mean (secs)**	**BER=1E-5 Mean (secs)**	**Means Difference (secs)**	**95% Confidence Limit**			**Means Difference (%)**
1K	0.082	0.435	-0.3529	-1.0399	0.3342		--
10K	3.829	5.340	-1.512	-7.343	4.320		--
100K	23.929	51.649	-27.720	-34.789	-20.652	*	-53.7%
1000K	207.292	567.325	-360.033	-377.140	-342.925	*	-63.5%

4.3.2.2 Asymmetric Channel Rate

The linear regression model using BER, link delay and file size as regressors is built for SCPS-VJ over asymmetric channel rate below:

$$\log_{10}(Time) = -4.08 - 40545.26 \times BER + 1.06 \times \log_{10}(FS) - 0.00016 \times Delay$$

$$+0.00006 \times Delay \times \log_{10}(FS) + 16.0 \times BER \times Delay + 10849.91 \times BER \times \log_{10}(FS)$$

When we compare SCPS-VJ models for both symmetric and asymmetric channel rates, we see that different from TCP/IP models, BER and link delay contribute more significantly to file transfer time in symmetric model than in asymmetric model for SCPS-VJ. This may be considered as one of key differences between SCPS-VJ and TCP/IP to cope with the problems of higher BER, long delay and asymmetric channels for space communication. This behavior is also expected for SCPS-Vegas.

Delay of 120 ms

Figure 4.37 shows that when asymmetric channel is used with a delay of 120 ms, the performance averages and regression lines do not bind together anymore and the performance difference is shown up for large files. This is similar to TCP/IP. Table 4.25 indicates that BER=1E-6 and BER=1E-5 have difference for 100K file and all three BERs have difference for 1000K file.

Delay of 1280 ms

Similar to TCP/IP, when the delay is increased to 1280 ms, the performance difference gets larger and the throughput differences are significant for large files as shown in Figure 4.38 and Table 4.26.

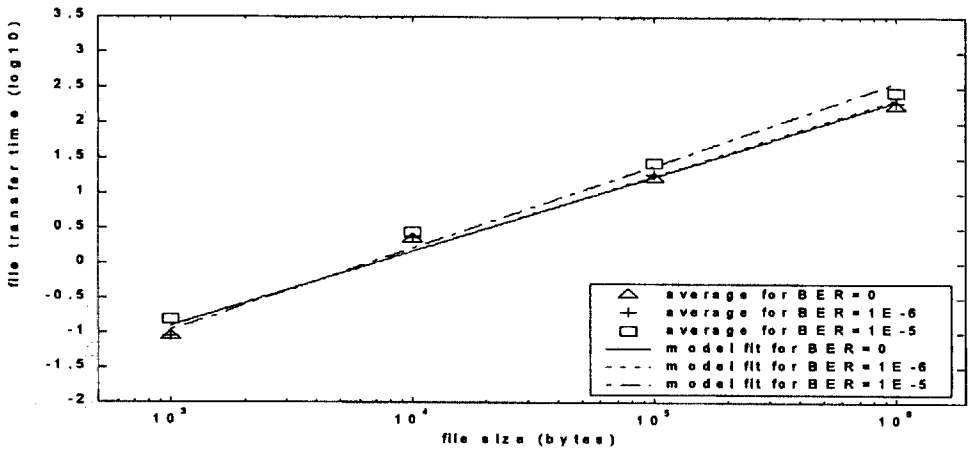

Figure 4.37: SCPS-VJ performance for different BERs over asymmetric channel with a delay of 120 ms

Table 4.25: SCPS-VJ comparison of means for different BERs over asymmetric channel with a delay of 120 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.089	0.088	0.0011	-0.6859	0.6881		--
10K	2.255	2.350	-0.095	-5.927	5.737		--
100K	17.022	19.132	-2.110	-9.178	4.958		--
1000K	174.482	192.331	-17.849	-34.956	-0.741	*	-9.3%
File Size	BER=1E-6 Mean (secs)	BER=1E-5 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.088	0.158	-0.0701	-0.7572	0.6169		--
10K	2.350	2.840	-0.489	-6.321	5.342		--
100K	19.132	27.596	-8.464	-15.532	-1.395	*	-30.7%
1000K	192.331	274.758	-82.427	-99.534	-65.320	*	-30%

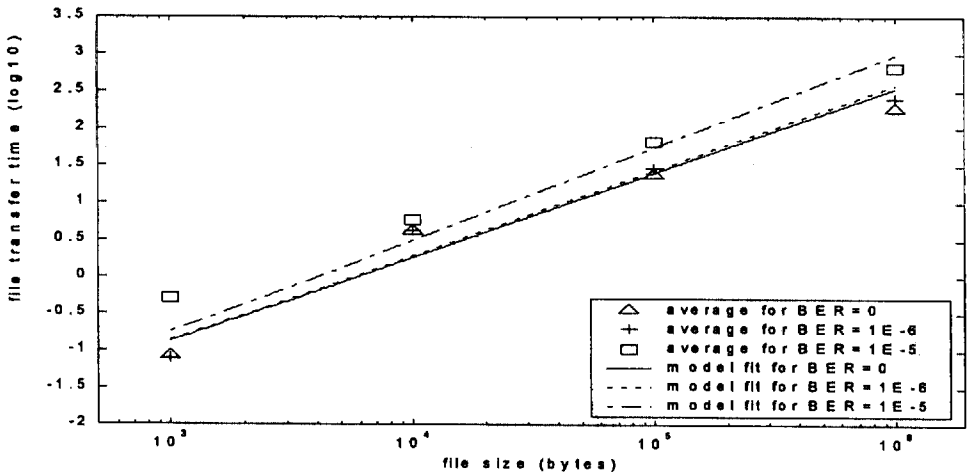

Figure 4.38: SCPS-VJ performance for different BERs over asymmetric channel with a delay of 1280 ms

Table 4.26: SCPS-VJ comparison of means for different BERs over asymmetric channel with a delay of 1280 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.086	0.081	0.0049	-0.6821	0.6919		--
10K	4.231	4.240	-0.009	-5.841	5.823		--
100K	24.205	29.338	-5.134	-12.202	1.935		--
1000K	190.625	250.517	-59.892	-76.999	-42.785	*	-23.9%
File Size	BER=1E-6 Mean (secs)	BER=1E-5 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.081	0.506	-0.4251	-1.1121	0.2619		--
10K	4.240	6.046	-1.805	-7.637	4.027		--
100K	29.338	65.687	-36.349	-43.417	-29.281	*	-55.3%
1000K	250.517	669.597	-419.080	-436.187	-401.973	*	-62.6%

4.3.3 Bit-Error-Rate Effects on SCPS-Vegas

4.3.3.1 Symmetric Channel Rate

The SCPS-Vegas linear regression model using BER, link delay and file size as regressors is built over symmetric channel rate below:

$$\log_{10}(Time) = -4.082 + 7350.09 \times BER + 1.014 \times \log_{10}(FS) + 0.00013 \times Delay$$
$$+0.00002 \times Delay \times \log_{10}(FS) + 9.538 \times BER \times Delay + 212.161 \times BER \times \log_{10}(FS)$$

Delay of 120 ms

Similar to SCPS-VJ, the performance points and regression lines bind tightly and tend to have linear relationship among BERs as shown in Figure 4.39. Table 4.27 shows that the only significant performance difference is between BER=1E-6 and BER=1E-5 for 1000K file. This is the same as TCP/IP and SCPS-VJ.

Delay of 1280 ms

When the delay of 1280 ms is used, SCPS-Vegas shows the same performance as TCP/IP and SCPS-VJ where the BERs show significant performance difference for large files. This can be seen from both Figure 4.40 and its corresponding HSD procedure Table 4.28. The models do not fit well for small files, which is surely related to the fact that a small file size doesn't show real protocol performance.

When we compare all BER effects of error free, 1E-6 and 1E-5, we find that SCPS-Vegas is insensitive to the increase of BER for symmetric channel. We expect that SCPS-Vegas should show similar BER effect for asymmetric channel based on the description of the congestion control mechanisms that SCPS-Vegas uses.

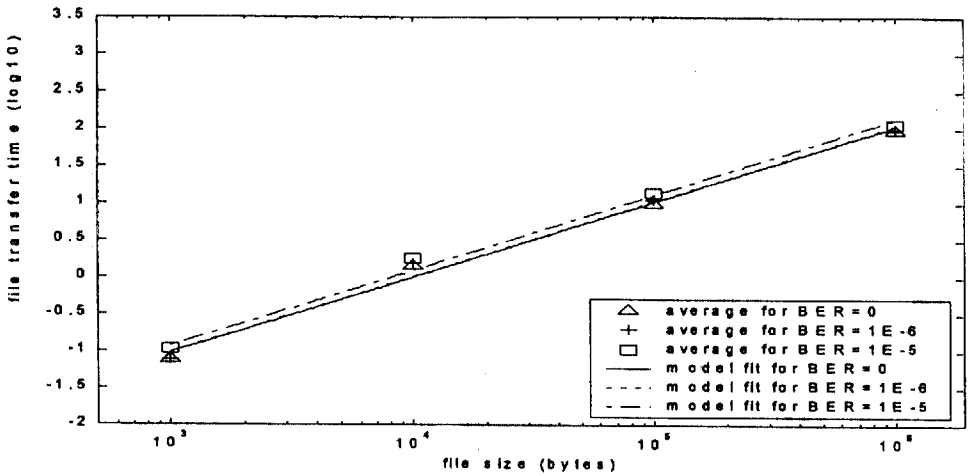

Figure 4.39: SCPS-Vegas performance for different BERs over symmetric channel with a delay of 120 ms

Table 4.27: SCPS-Vegas comparison of means for different BERs over symmetric channel with a delay of 120 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.078	0.078	-0.0005	-0.6835	0.6826		--
10K	1.442	1.568	-0.1254	-1.2380	0.9872		--
100K	10.190	11.052	-0.862	-4.739	3.016		--
1000K	95.351	97.402	-2.051	-11.028	6.925		--
File Size	BER=1E-6 Mean (secs)	BER=1E-5 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.078	0.109	-0.0301	-0.7131	0.6529		--
10K	1.568	1.789	-0.2218	-1.3344	0.8909		--
100K	11.052	14.006	-2.955	-6.832	0.923		--
1000K	97.402	117.240	-19.838	-28.814	-10.862	*	-16.9%

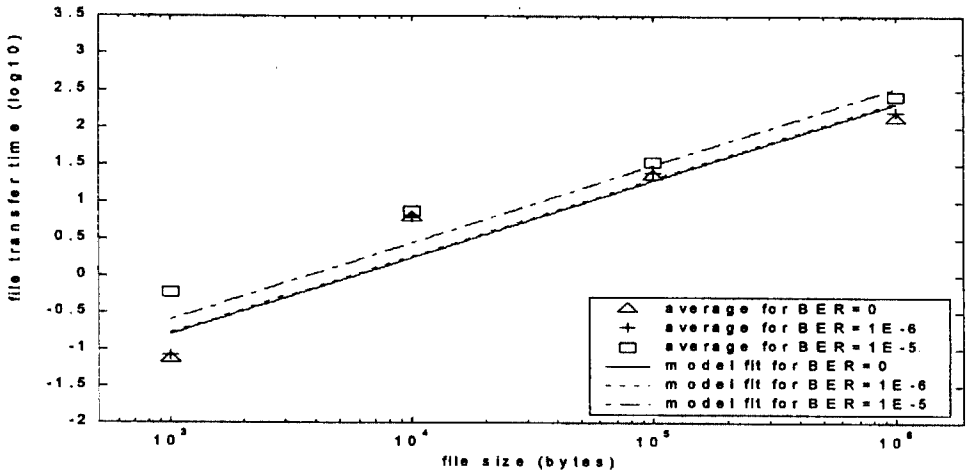

Figure 4.40: SCPS-Vegas performance for different BERs over symmetric channel with a delay of 1280 ms

Table 4.28: SCPS-Vegas comparison of means for different BERs over symmetric channel with a delay of 1280 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.073	0.084	-0.0110	-0.6940	0.6720		--
10K	6.074	6.086	-0.0116	-1.1243	1.1010		--
100K	22.675	24.102	-1.427	-5.304	2.451		--
1000K	135.183	162.009	-26.826	-35.803	-17.850	*	-16.6%
File Size	BER=1E-6 Mean (secs)	BER=1E-5 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.084	0.603	-0.5183	-1.2013	0.1647		--
10K	6.086	7.351	-1.2651	-2.3778	-0.1525	*	-17.2%
100K	24.102	33.944	-9.842	-13.720	-5.965	*	-29%
1000K	162.009	267.870	-105.861	-114.837	-96.884	*	-39.5%

4.3.3.2 Asymmetric Channel Rate

As mentioned when we studied the BER effect on SCPS-Vegas over symmetric channel rate, we expect SCPS-Vegas to be insensitive to the increase of BER over asymmetric channel rate. The following linear regression model reflects SCPS-Vegas's linear relationship between the file transfer time and the transmission condition over asymmetric channel rate:

$$\log_{10}(Time) = -4.035 - 7932.12 \times BER + 1.059 \times \log_{10}(FS) - 0.00004 \times Delay$$
$$+0.00003 \times Delay \times \log_{10}(FS) + 2.635 \times BER \times Delay + 4240.98 \times BER \times \log_{10}(FS)$$

When we compare the models for both symmetric and asymmetric channels for SCPS-Vegas, we see that BER, link delay and their interaction contribute more significantly for symmetric channel than for asymmetric channel. Different from all models for TCP/IP and SCPS-VJ, the huge magnitude difference of BER coefficients between symmetric and asymmetric channel models for SCPS-Vegas might make SCPS-Vegas insensitive to channel rate change from symmetric to asymmetric, especially if 1280 ms delay is used since, as mentioned, the interaction between BER and link delay contributes less significantly for asymmetric channel rate. This is to be expected when we conduct our analysis for asymmetric channel rate.

Delay of 120 ms

Similar to TCP/IP, when the asymmetric channel is used with a delay of 120 ms, the performance difference among all BERs get significant along the increase of the file size as shown in Figure 4.41 and Table 4.29. We expect this difference to be more significant if the delay of 1280 is used.

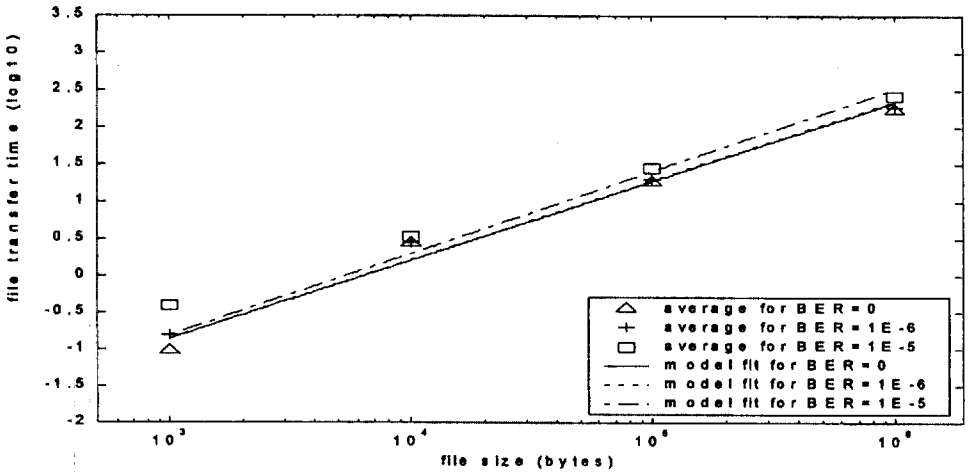

Figure 4.41: SCPS-Vegas performance for different BERs over asymmetric channel with a delay of 120 ms

Table 4.29: SCPS-Vegas comparison of means for different BERs over asymmetric channel with a delay of 120 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.095	0.159	-0.0643	-0.7473	0.6187		--
10K	2.814	2.867	-0.0536	-1.1663	1.0590		--
100K	18.976	20.027	-1.050	-4.928	2.827		--
1000K	177.758	189.916	-12.158	-21.134	-3.182	*	-6.4%
File Size	BER=1E-6 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.159	0.396	-0.2363	-0.9193	0.4467		--
10K	2.867	3.422	-0.5542	-1.6668	0.5585		--
100K	20.027	28.595	-8.568	-12.446	-4.691	*	-30%
1000K	189.916	261.204	-71.287	-80.264	-62.311	*	-27.3%

Delay of 1280 ms

In the same way as TCP/IP, when the delay time is increased, the BERs tend to show significant performance difference for large files as shown in Figure 4.42. Table 4.30 lists pairs showing significant performance difference ranging from 10K to 1000K. When we compare figures and tables between symmetric channel and asymmetric channel, we see the figures are very identical and the performance differences are very close. This imply that reducing symmetric forward channel to 2400 bps does not significantly affect the performance relationship among BERS when the channel has a delay of 1280 ms. This is different from TCP/IP and SCPS-VJ which show that asymmetric channel mostly has much large performance difference. This verifies our prediction based on the model at the beginning. In summary, we have the following conditions for the BER effect on the performance of SCPS-Vegas:

- Similar to TCP/IP and SCPS-Vegas, the increase of BER from error free to 1E-6 does not significantly affect SCPS-Vegas's performance when relative small files are transmited with symmetric channel and 120 ms delay. BER=1E-5 decreases the throughput of SCPS-Vegas much more seriously.
- SCPS-Vegas is insensitive to the increase of BER for both symmetric and asymmetric channel rates. Reducing channel rate from symmetric to asymmetric does not affect the SCPS-Vegas performance relationship among BERs if a delay of 1280 is used. This can be understood from that SCPS-Vegas is developed to copy with the problems of asymmetric channel rate, high BERs and long link delays in space.

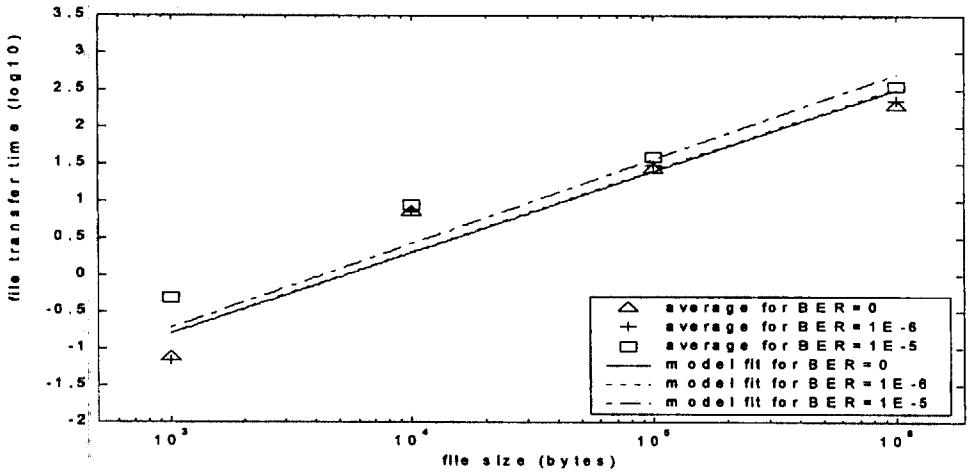

Figure 4.42: SCPS-Vegas performance for different BERs over asymmetric channel with a delay of 1280 ms

Table 4.30: SCPS-Vegas comparison of means for different BERs over asymmetric channel with a delay of 1280 ms

File Size	BER=0 Mean (secs)	BER=1E-6 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.078	0.068	0.0098	-0.6732	0.6928		--
10K	7.199	7.245	-0.0451	-1.1577	1.0675		--
100K	27.754	29.958	-2.203	-6.081	1.674		--
1000K	193.112	223.997	-30.885	-39.862	-21.909	*	-13.8%
File Size	BER=1E-6 Mean (secs)	BER=1E-5 Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.068	0.495	-0.4268	-1.1098	0.2562		--
10K	7.245	8.872	-1.6272	-2.7398	-0.5145	*	-18.3%
100K	29.958	40.146	-10.188	-14.065	-6.311	*	-25.4%
1000K	223.997	348.515	-124.518	-133.494	-115.542	*	-35.7%

Conclusions

Combining the above effect of BER on each of three protocols, we may conclude:

- When relative small files (< 1Mbytes) were transmitted with symmetric channel rate 115,200 bps:115,200 bps, the increase of BER from error free to 1E-6 does not statistical significantly affect the protocols' performance.
- When a BER of 1E-5 is used for both channel rates, the decrease of throughput is seriously for all TCP/IP, SCPS-VEGAS and SCPS-Vegas.
- The joint conditions of a large file, a long delay makes all protocols show statistically significant performance difference with respect to the change of BER. Based on this result, we reject all **Hypotheses Set 1** through **Hypotheses Set 6** corresponding to the analysis set (3) in Section 3.2.1 and conclude that no pair protocols have equal file transfer time means with each of BER changes from 0 to 1E-6 and from 1E-6 to 1E-5 for each of the same transmission conditions of channel rate, link delay and file size. Similar to rejecting hypotheses for protocol performance comparisons in Section 4.1 and for delay effects study in Section 4.2, this does not mean that each of three protocols has significantly different file transfer time means with each of BER changes from 0 to 1E-6 and from 1E-6 to 1E-5 for each of the same transmission conditions of channel rate, link delay and file size.
- The factors of file size, BER and link delay and all their interactions contribute more significantly to the protocol performance.

- TCP/IP is very sensitive to the increase of BER and SCPS-VJ is relatively sensitive to the increase of BER.
- SCPS-Vegas is insensitive to the increase of BER for both symmetric and asymmetric channel rates.
- Reducing the forward channel rate from 115,200 bps to 2400 bps does not significantly affect the SCPS-Vegas performance relationship corresponding to the change of BER if the link delay of 1280 is used.
- The above conclusion partially addresses our basic question (3) on how BER affects each protocol performance listed in Chapter 1.

In summary, to have a qualitative and qua knowledge of the space channel effects on each protocol performance, Table 4.31 provides mean differences of the overall averaged transmission times in terms of channel condition changes. Based on these quantitative mean differences, the effects of the transmission condition changes on protocols' performance are also estimated in a qualitative form.

When we compare Table 4.31 with our expected qualitative effects in Table 2.1, we see they match very well. This tells us that our qualitative effect prediction based on different features of each protocol is basically accurate.

Table 4.31: Effects of space channel conditions on each protocol performance

		TCP/IP		SCPS-VJ		SCPS-Vegas	
		Mean Difference (second)	Qualitative Effects	Mean Difference (second)	Qualitative Effects	Mean Difference (second)	Qualitative Effects
Link Delay Effects	120ms ↓ 1280ms	-78.39	Highly Sensitive	-49.06	Moderately Sensitive	-20.71	Slightly Sensitive
Channel Rate Effects	SYM ↓ ASYM	-40.04	Moderately Sensitive	-23.54	Slightly Sensitive	-24.05	Slightly Sensitive
Bit-Error-Rate (BER) Effects	0 ↓ 1E-6	-18.36	Highly Sensitive	-10.19	Moderately Sensitive	-4.86	Slightly Sensitive
	0 ↓ 1E-5	-104.40		-71.60		-27.23	

5 BEHAVIOR OF TCP/IP AND SCPS OVER A SATELLITE LINK

Figure 5.1 outlines test factors and different levels of each factor for experiment over satellite link. The tests over realistic satellite link are expected to bring three benefits: (1) Extend performance to cover large BDP region as well; (2) Compare the performance of protocols in practical sense; (2) Improve the SGLS test-bed based on the analysis of test results.

Due to the restriction of available satellite link conditions, the experiments over the satellite link are done only with a delay of 120 ms and error free. The channel baud rates include: (1) 115,200 bps:115,200 bps; (2) 4 Mbps:4 Mbps and (3) 4 Mbps:57,600 bps. With rate (1), we expect to validate the SGLS test-bed performance by comparing the in-lab results with the actual satellite channel results under the conditions of slow symmetric channel rate. With rates (2) and (3), we aim to extend the performance analysis to cover large BDP region. Appendices D provides the mean and standard deviation of file transfer time of 16 observations for each experimental treatment over satellite link.

As in Chapter 4, based on the relationship between the protocols and the relationship between two congestion control modes of SCPS protocol, the analysis using SAS based HSD procedures will be done for each of the following protocol comparison sets:

(1) TCP/IP versus SCPS-VJ;

(2) SCPS-VJ versus SCPS-Vegas.

Each of the above two comparison sets has 6 configurations (=2 Protocol × 3 Channel Rate). The number of observations is 96 (=16 × 6) since there are 16 observations for each treatment.

Section 5.1 concentrates on comparing the performance of two protocols over satellite link with each of the above three baud rates. Section 5.2 analyzes the protocol performance difference between the SGLS test-bed and satellite link with rate (1) for each protocol. The analysis will also be primarily supported using the SAS based HSD procedure.

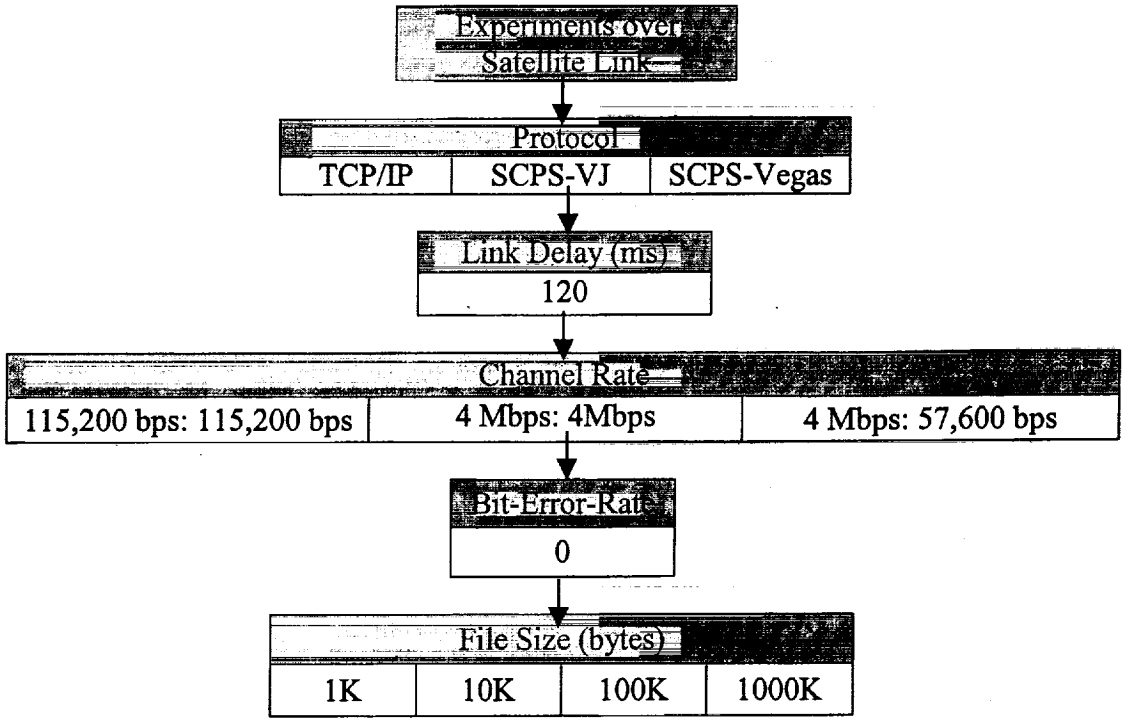

Figure 5.1: Test factors and different levels of each factor for experiments over satellite link

5.1 Comparing the Performance of TCP/IP and SCPS

This section concentrates on comparing the protocol performance over satellite link with each of the following three baud rates: (1) 115,200 bps:115,200 bps; (2) 4 Mbps:4 Mbps and (3) 4 Mbps:57,600 bps. For our experiments over satellite link, rate (1) is considered to be slow symmetric channel rate and rate (2) and rate (3) to be high speed symmetric rate and high speed asymmetric rate respectively. The above experiments over satellite link are conducted only with a delay of 120 ms and error free due to the restriction of available satellite link conditions. The protocol performance for transmitting 1000Kbytes file with channel rate 4 Mbps:57,600 bps is also analyzed using performance graphs to examine the protocol behavior in detail. Considering the protocols' inconsistent performance caused by the experimental configuration changes (e.g., test hardware change and computer system upgrade) on the ground station, the linear regression models are not built for the regression of the protocol performance on the transmission conditions over satellite link.

Slow Symmetric Channel Rate of 115,200 bps:115,200 bps

Figure 5.2 plots the protocol performance over satellite link with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms. Close to the protocol plot over the SGLS test-bed in Figure 4.1, we see three protocol are basically very close, especially that TCP/IP and SCPS-VJ even can not be distinguished. The only point with performance difference locates between SCPS-Vegas and other two for 10K file. The HSD procedure in Table 5.1 verifies the above observation. By checking the original time report for each run, the above significant performance difference

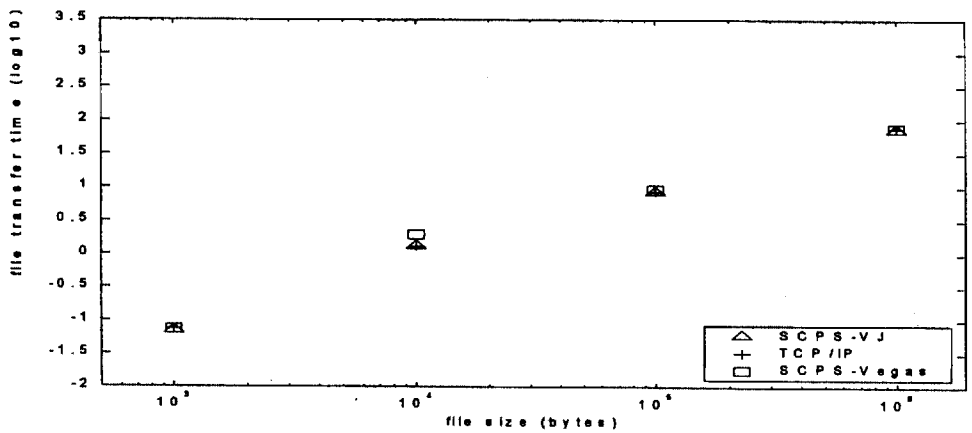

Figure 5.2: Satellite time for TCP/IP, SCPS-VJ and SCSP-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

Table 5.1: Satellite comparison of means for channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.075	0.073	0.002703	-0.013620	0.019026		--
10K	1.307	1.337	-0.03039	-0.09236	0.03158		--
100K	8.750	8.486	0.2637	-0.6687	1.1961		--
1000K	75.90	75.687	0.2132	-0.5623	0.9886		--
File Size	**SCPS-Vegas Mean (secs)**	**SCPS-VJ Mean (secs)**	**Means Difference (secs)**	**95% Confidence Limit**			**Means Difference (%)**
1K	0.074	0.073	0.001253	-0.01751	0.02002		--
10K	1.942	1.337	0.60512	0.51178	0.69845	*	45.3%
100K	9.180	8.486	0.6932	-0.2701	1.6566		--
1000K	76.417	75.687	0.7303	-0.3607	1.8214		--

consistently comes from all 16 runs instead of any exceptional runs. Similar to the experiments over the SGLS test-bed, this difference actually comes from the fundamental behavior difference between VJ's traditional Slow Start mechanism and Vegas's modified Slow Start mechanism as we explained in Section 4.1.1.1 for link delay of 1280 ms.

High Speed Symmetric Channel Rate 4 Mbps:4 Mbps

The protocol performance over satellite link with channel rate of 4 Mbps:4 Mbps, BER=0 and Delay=120 ms is plotted in Figure 5.3. Different from the plot for channel rate of 115,200 bps:115,200 bps, we see each of four files in Figure 5.3 takes much less time than their corresponding file in Figure 5.2 does. Clearly, this great improvement is due to the effect of high speed symmetric channel rate 4 Mbps:4 Mbps, which is about thirty-four times faster than 115,200 bps:115,200 bps is. By observing the plot, we expect to see the performance difference for large file points where SCPS-VJ provides the highest throughput while the performance of SCPS-Vegas is clearly poorer than other two. The HSD procedure in Table 5.2 indicates the performance of all three protocols are significantly different each other. Similar to 1K file mean difference between SCPS-Vegas and SCPS-VJ with channel rate 115,200 bps:115,200 bps, since all the means are very small and the Mean Differences(%) are relatively small, these pairs may not have practically significant differences.

High Speed Asymmetric Channel Rate 4 Mbps:57,600 bps

Figure 5.4 plots the protocol performance over high speed asymmetric satellite link with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms. The direct

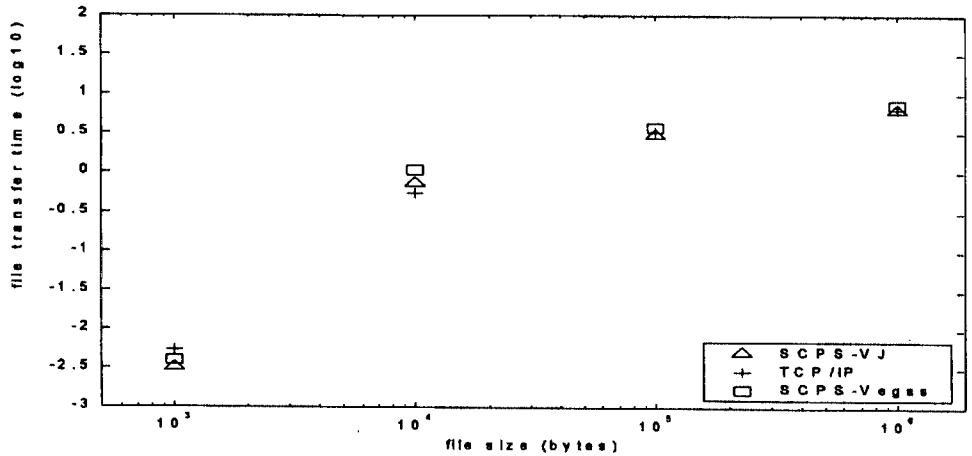

Figure 5.3: Satellite time for TCP/IP, SCPS-VJ and SCSP-Vegas with channel rate of 4 Mbps:4 Mbps, BER=0 and Delay=120 ms

Table 5.2: Satellite comparison of means for channel rate of 4 Mbps:4 Mbps, BER=0 and Delay=120 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.005	0.003	0.002097	-0.003481	0.00767		--
10K	0.555	0.741	-0.186488	-0.195729	-0.17724	*	-25.2%
100K	3.282	3.066	0.21632	0.17773	0.25490	*	7.1%
1000K	6.591	6.386	0.20481	0.15693	0.25270	*	3.2%
File Size	SCPS-Vegas Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.004	0.003	0.000714	-0.003635	0.00506		--
10K	1.093	0.741	0.352326	0.342933	0.36171	*	47.5%
100K	3.833	3.066	0.76758	0.72929	0.80587	*	25%
1000K	7.491	6.386	1.10540	1.01107	1.19973	*	17.3%

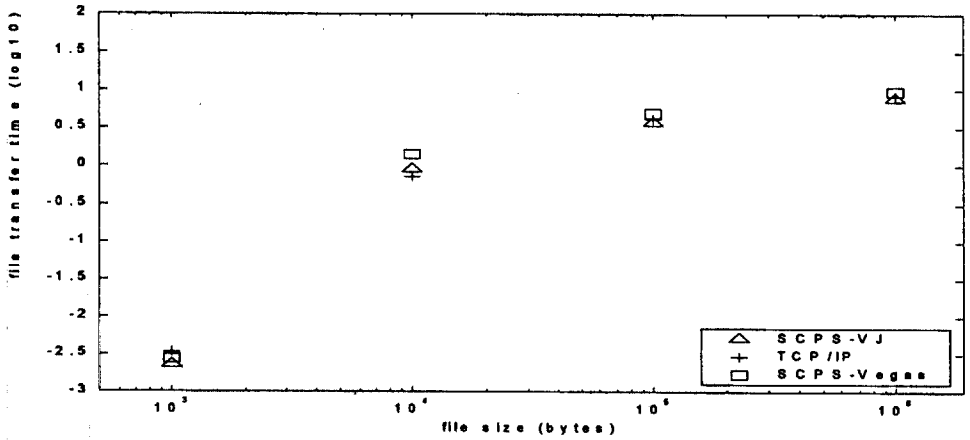

Figure 5.4: Satellite time for TCP/IP, SCPS-VJ and SCSP-Vegas with channel rate of 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

Table 5.3: Satellite comparison of means for channel rate of 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

File Size	TCP Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.003	0.002	0.001007	-0.004571	0.00658		--
10K	0.718	0.913	-0.195512	-0.204753	-0.18627	*	-21.4%
100K	4.246	3.766	0.48007	0.44148	0.51866	*	12.7%
1000K	8.474	7.892	0.58158	0.53370	0.62946	*	7.4%
File Size	SCPS-Vegas Mean (secs)	SCPS-VJ Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.003	0.002	0.000358	-0.003991	0.004708		--
10K	1.402	0.913	0.489276	0.479883	0.498669	*	53.6%
100K	4.915	3.766	1.14897	1.11068	1.18726	*	30.5%
1000K	9.631	7.892	1.73862	1.64429	1.83295	*	22%

observation is that the performance tendency is very similar to that for symmetric link with rate 4 Mbps:4 Mbps. This is shown up from both each protocol performance for all files and the performance relationship among all protocols. Comparing to Figure 5.3, we also see that all protocols show slight lower throughput for each of four files. This is basically caused by slower speed acknowledge link with rate 57,600 bps, which cannot fully support 4 Mbps data transmission. We see statistically significant performance difference for large files can also be seen from the comparisons of means provided by Table 5.3. Similar to the comparisons with channel rate 4 Mbps:4 Mbps, considering very small mean values for all pairs with even smaller mean differences, they may not have practically significant differences.

When we compare the protocol performance comparison results between channel rate 4 Mpbs:57,600 bps and asymmetric low BDP link in our test-bed (i.e., 115,200 bps:2400 bps in Section 4.1.2.1) with BER=0 and Delay=120 ms, we see both results show that all three protocols have no "practically" significant performance differences. This tells us that the overall high channel rates do not affect the comparison results between protocols too much and test results from different test environments match each other. In order to understand how three protocols perform differently in detail, let's analyze their behavior at 1000K point in Figure 5.4, i.e., their behavior for transmitting 1000 Kbytes file with rate 4 Mpbs:57,600 bps, BER=0 and Delay=120 ms. The connection for each protocol is chosen randomly from 16 runs of each protocol. We will use various graphs obtained by using network analysis tools to conduct the analysis. These graphs are briefly described in Section 3.1.5.3.

Table 5.4: A portion of transmission data of TCP/IP, SCPS-VJ and SCPS-Vegas for transmitting 1000 Kbytes data with rate of 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

	TCP/IP	SCPS-VJ	SCPS-Vegas
Data Transmission Time (secs)	8.399	7.866	9.626
Elapsed Time (secs)	9.903	9.520	11.127
Throughput (bps)	100,977	105,043	89,871
Average Advertised Window (bytes)	279,108	560,078	560,078
Average Congestion Window (bytes)	165,098	168,908	145,310
Average RTT (ms)	771.7	780.2	764.2
RTT Stdev (ms)	56.7	70.8	61.9

Table 5.4 lists some of transmission statistical data for each protocol. Those data will be used to support the analysis.

Figure 5.5 plots the time sequence numbers with respective to file transfer time for three protocols. We see three protocols basically start at the same point and finally end up at different positions. The sequence number plots for three protocols starts showing separations around 5.00 seconds, which is about in the middle of the transmissions, and the difference seems to get larger and larger along with the file size increase. This matches the results of the comparisons of means in Table 5.4. Figure 5.6 is the "zoom in" of the end portion of the time sequence graph in Figure 5.5. Figure 5.6 clearly shows that SCPS-VJ takes the least time to transmit 1000 Kbytes file and TCP/IP is between SCPS-VJ and SCPS-Vegas. The data transmission time and the elapsed time for all protocols are given in Table 5.4. We see SCPS-VJ and TCP/IP are closer while SCPS-Vegas is far away from other two.

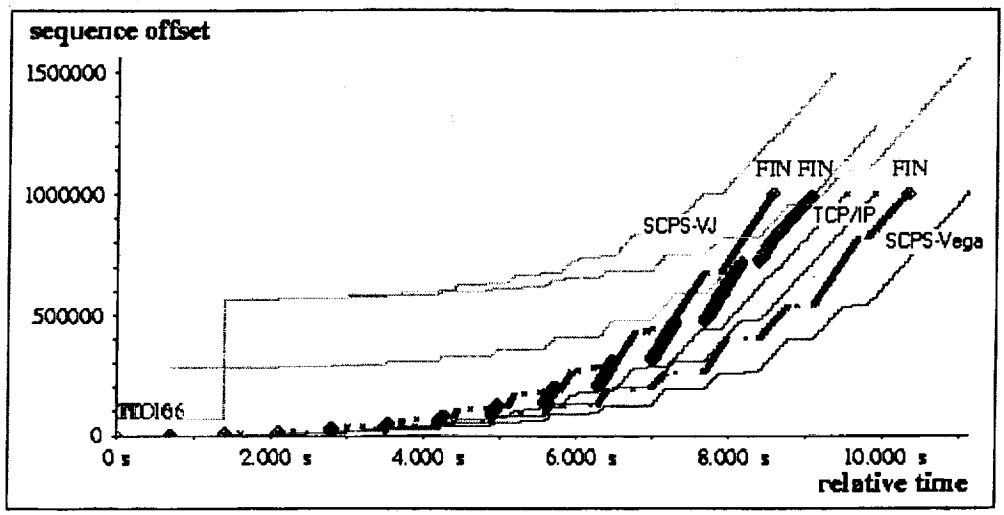

Figure 5.5: Time sequence numbers with respect to file transfer time for three protocols transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

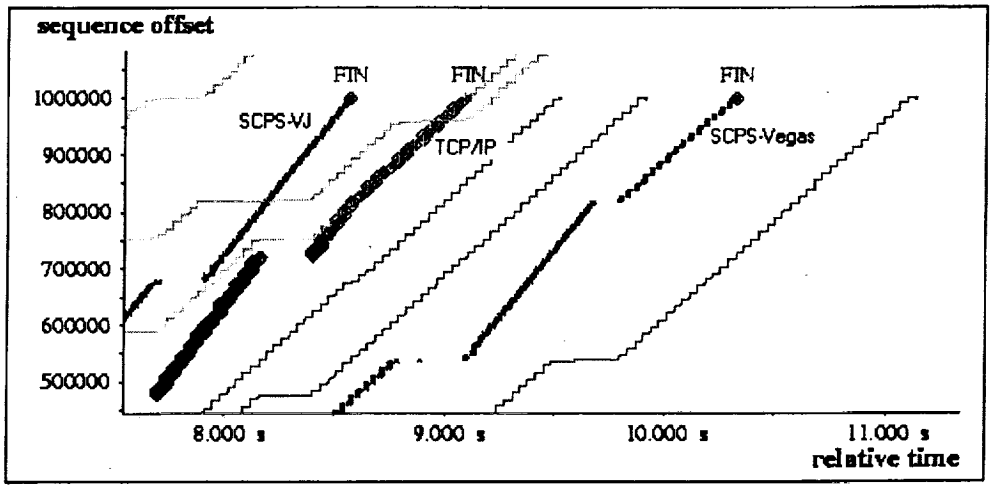

Figure 5.6: Enlarged view of the end portion of the time sequence plots in Figure 5.5

Figures 5.7-5.9 individually plot the time sequence numbers for each protocol. Three plots all show that the protocol starts from the initial window and increase the transfer rate using the "Slow Start." The "zoom in" versions of the time sequence plot for each protocol are shown in Figures 5.10-5.12.

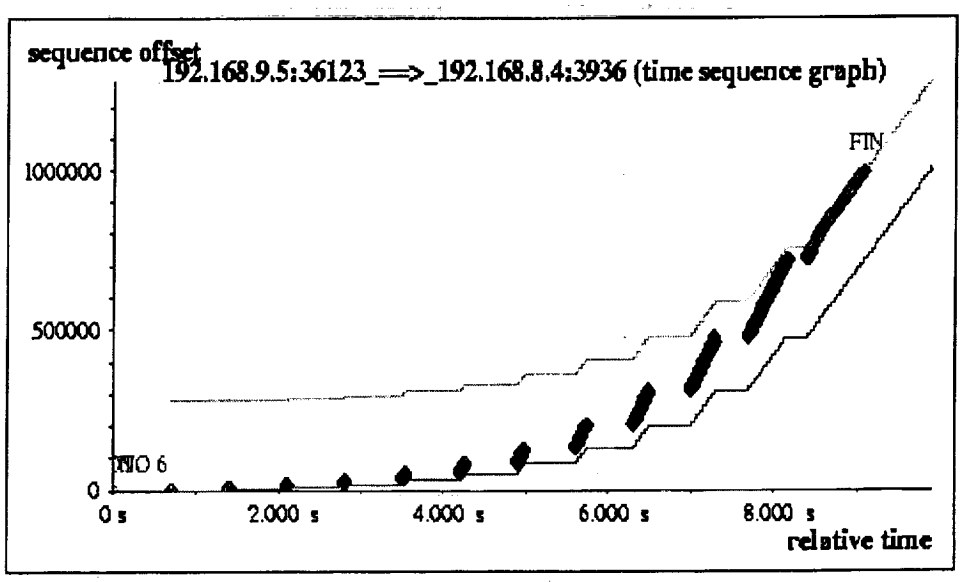

Figure 5.7: Time sequence numbers of TCP/IP transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

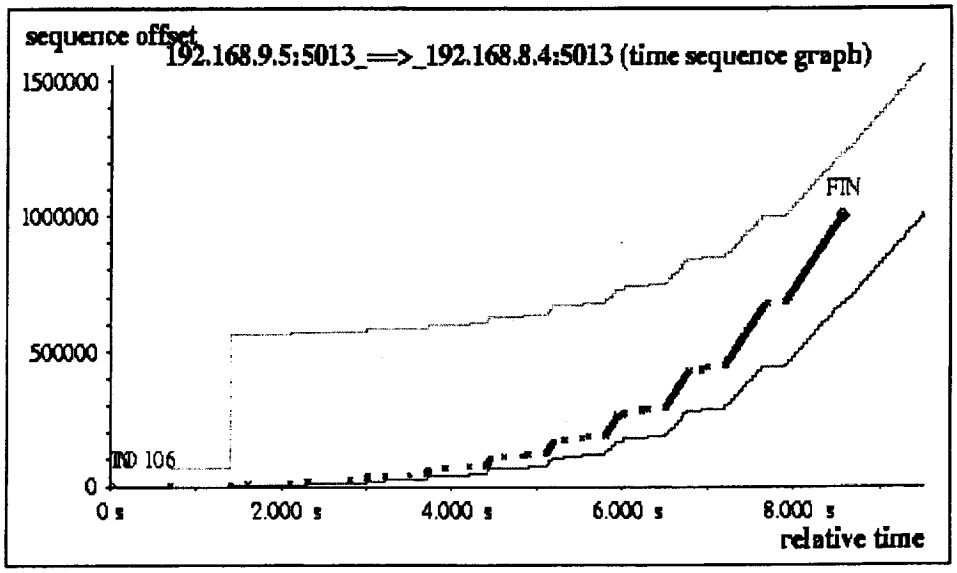

Figure 5.8: Time sequence numbers of SCPS-VJ for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

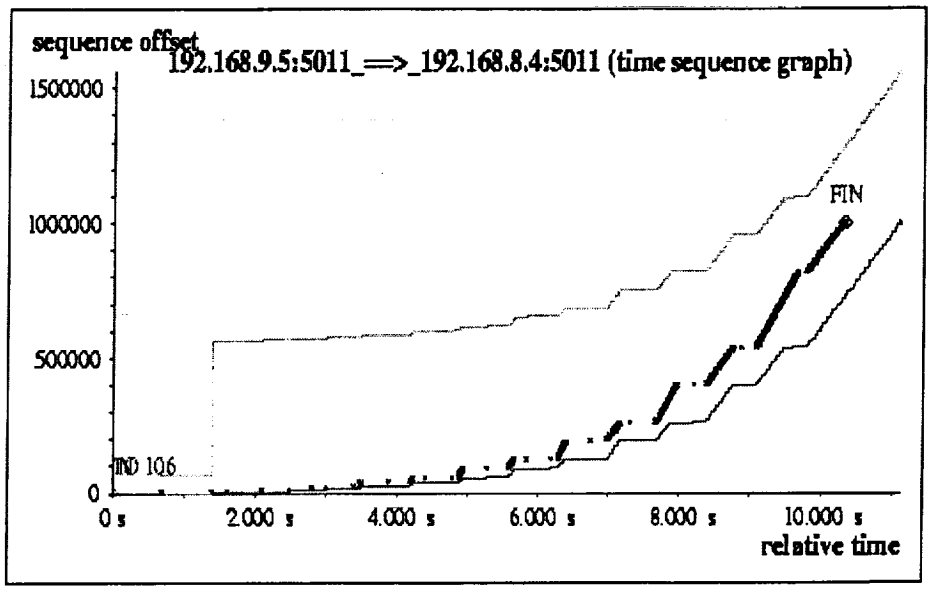

Figure 5.9: Time sequence numbers of SCPS-Vegas for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

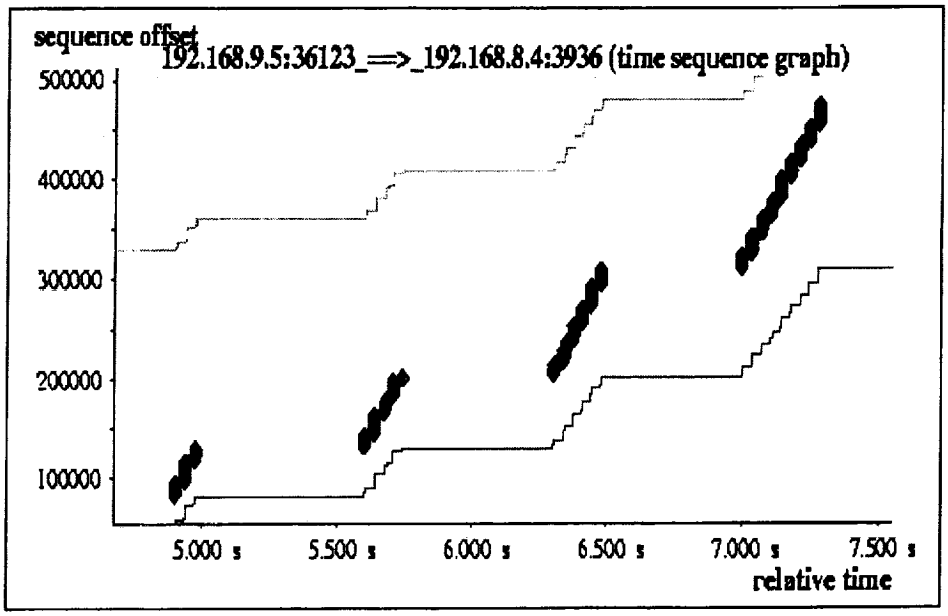

Figure 5.10: Enlarged view of time sequence graph of TCP/IP in Figure 5.7

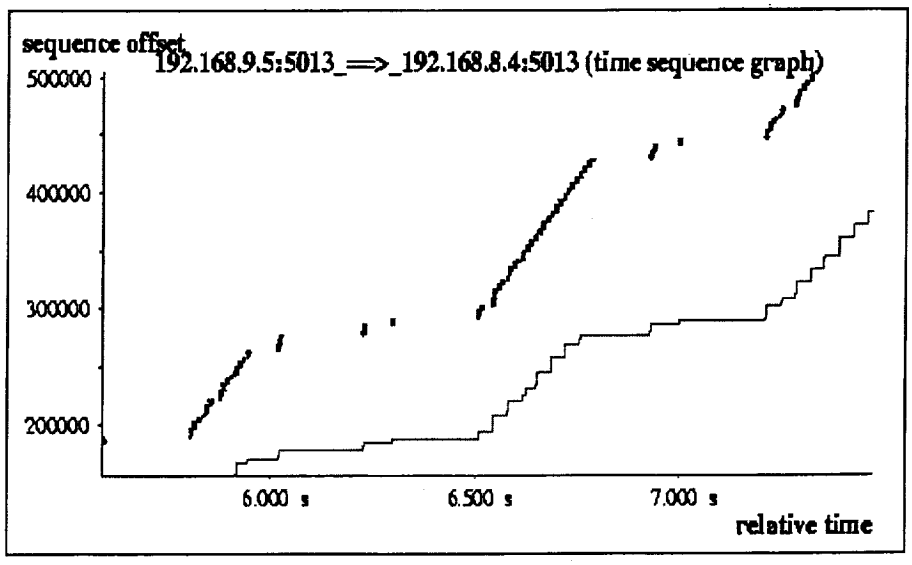

Figure 5.11: Enlarged view of time sequence graph of SCPS-VJ in Figure 5.8

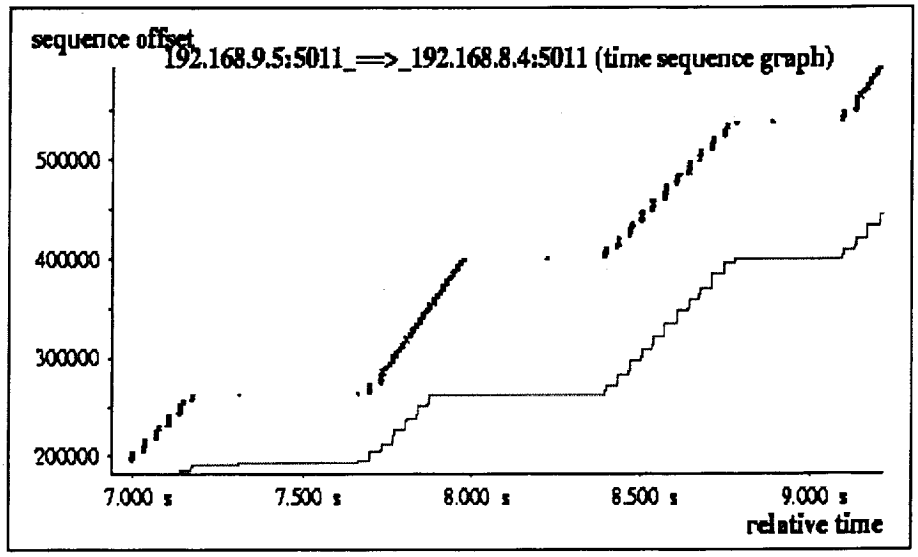

Figure 5.12: Enlarged view of time sequence graph of SCPS-Vegas in Figure 5.9

By looking at three plots in Figures 5.7-5.9, we see there are no retransmissions for all three protocols since BER=0 is used for the file transfers. Basically, all returning packets update both the acknowledgement line and the edge of

the window. The baud rate 4 Mbps:57,600 bps does not limit the data transfer. Instead, the transmitter tries to gradually "fill out" the pipe using the "Send and Wait for ACK" procedure, which increases the congestion window for each ACK received until all data is sent out as shown by the congestion window plot in Figure 5.13.

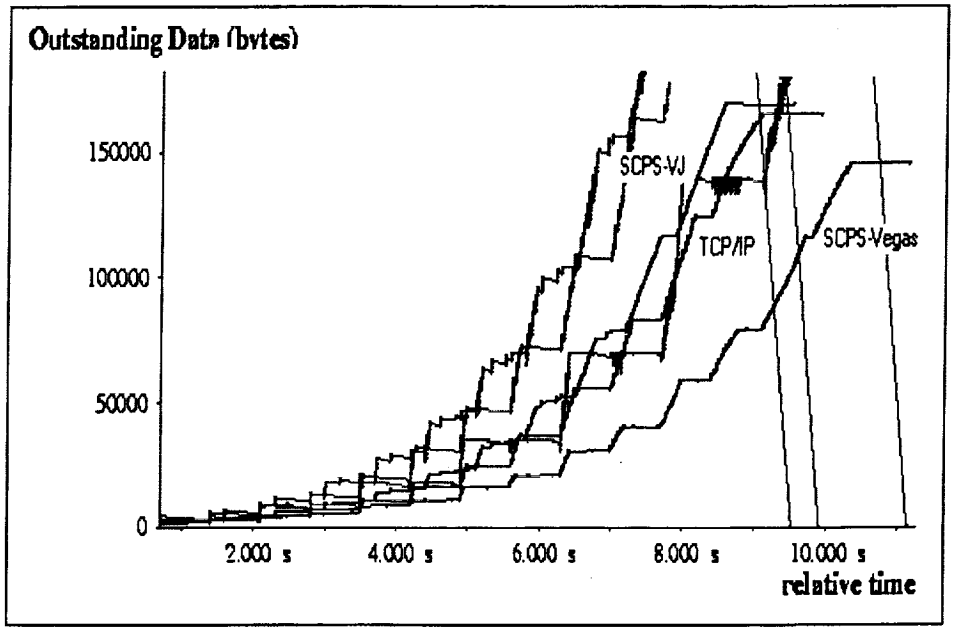

Figure 5.13: Congestion window graph of three protocols for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

A special case is that, for TCP/IP, as shown at the end of Figure 5.7, the advertised window is full along the last packet flight is being sent and thus starts to limit the data sending rate. This is happening with neither SCPS-VJ nor SCPS-Vegas since, if we compare the advertised window lines of TCP/IP and that of SCPS-VJ and SCPS-Vegas, we see the window size for both SCPS-VJ and SCPS-Vegas are advertised twice of that for TCP/IP. Table 5.4 shows the average advertised window

for TCP/IP is 279,108 bytes while it is 560,078 bytes for both SCPS-VJ and SCPS-Vegas. Such a big difference of window size seems to provide SCPS-VJ and SCPS-Vegas more space to support the high sending rate of 4 Mbps. The "Send and Wait for ACK" problem is basically caused by waiting for delayed slow link ACKs with rate 57,600 bps after each packet flight ends.

Along the increase of the packets for each flight which corresponds to the increase of the congestion window, the RTT for each flight is also increased gradually as shown by the RTT plots in Figures 5.14-5.16. This is particularly clear for TCP/IP RTT graph in Figure 5.14.

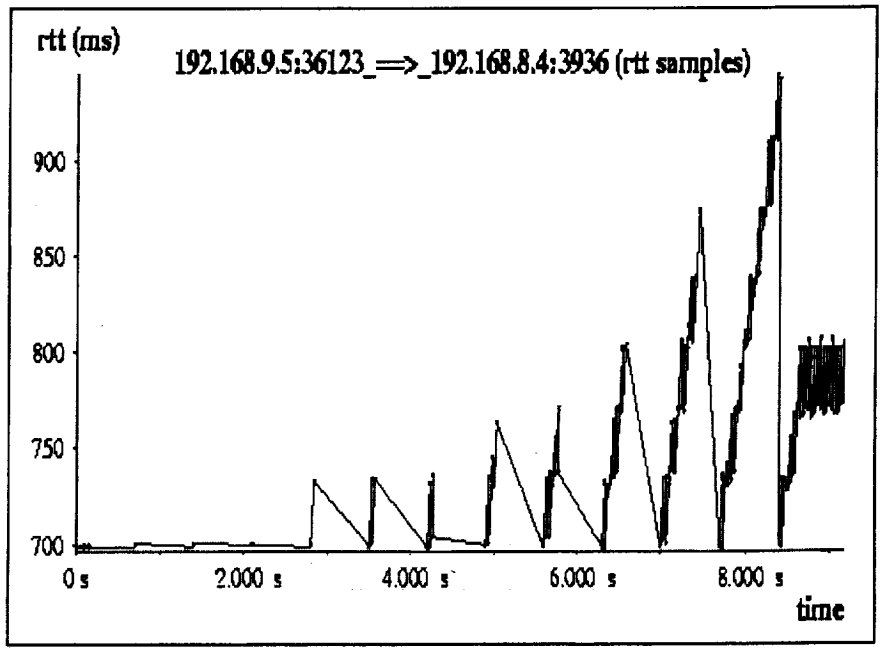

Figure 5.14: RTT graph of TCP/IP for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

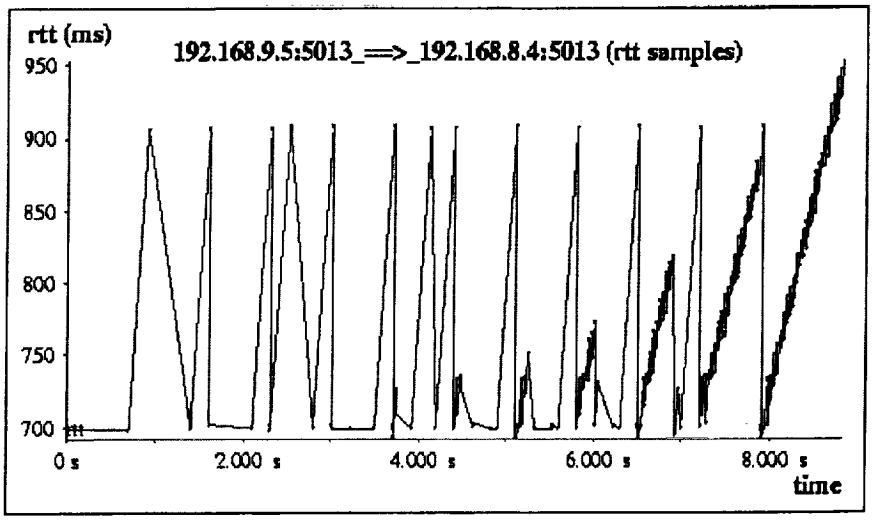

Figure 5.15: RTT graph of SCPS-VJ for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

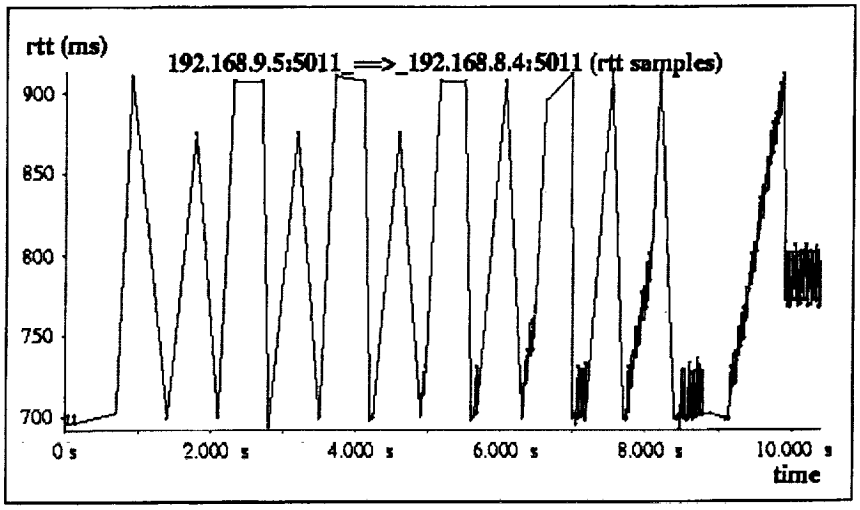

Figure 5.16: RTT graph of SCPS-Vegas for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

The strongly varied "saw-like" RTT portions for SCPS-VJ and SCPS-Vegas correspond to the limited number of packets sent between packet flights, and each big

packet flight corresponds to an approximately increased RTT segment. This can be observed clearly when we match Figure 5.8 and Figure 5.9 to Figure 5.15 and Figure 5.16. Table 5.4 shows that SCPS-VJ has the largest average RTT 780.2 ms while TCP/IP and SCPS-Vegas have the average RTT of 771.7 ms and 764.2 ms. The "saw-like" RTT is not happened for TCP/IP. This is to be expected since there are no packets injected into the pipe when the transmitter waits for receiving ACKs from the receiver as we see in Figure 5.7 and Figure 5.10.

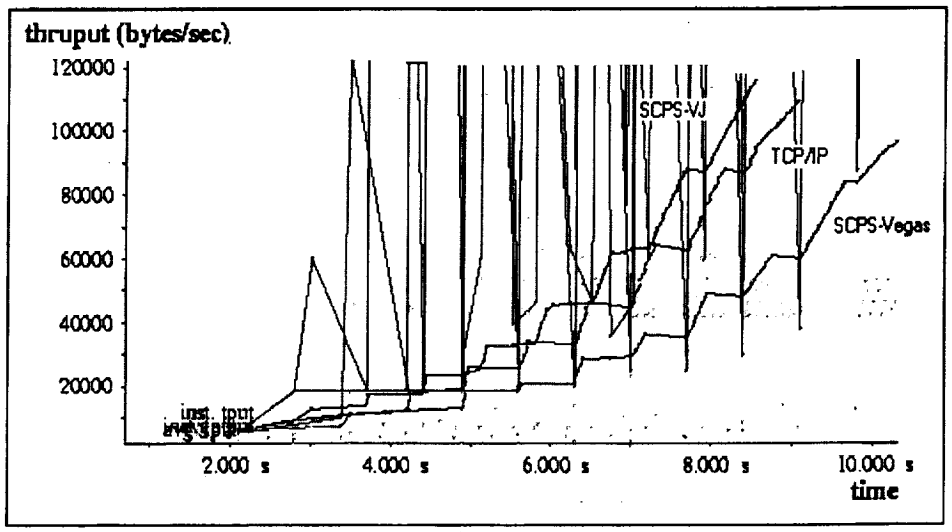

Figure 5.17: Throughput graph of three protocols for transmitting 1000 Kbytes data with rate 4 Mbps:57,600 bps, BER=0 and Delay=120 ms

We know the capacity of the pipe between the transmitter and the receiver can be calculated as

$$\text{Capacity (bits)} = \text{Bandwidth (bits/sec)} \times \text{RTT (sec)}$$

The capacity can vary widely depending on the network speed and the RTT between the two ends. Provided we have a fixed 4 Mbps bandwidth for all protocols,

the largest RTT of SCPS-VJ 780.2 ms brings us the highest average congestion window 168,908 bytes and the smallest RTT of SCPS-Vegas 764.2 ms gives us the smallest average congestion window 145, 310 bytes. This can be intuitively seen from the congestion window graph in Figure 5.13. Corresponding to the increase of the congestion window and the increase of RTT for each protocol, the averaged throughput for three protocols are plotted in Figure 5.17, which shows that SCPS-VJ has the highest throughput and SCPS-Vegas has the lowest one. The specific throughput for each protocol is provided in Table 5.4. We see all the throughputs are mostly limited by the long delayed slow-speed acknowledgements. The throughput for TCP/IP is also partially limited by the receiver's ability to keep the window open at the end of the connection as shown in Figure 5.7.

In the end, by observing the plots in Figures 5.7-5.9, we see along the increase of the congestion window, while the RTT gets larger and larger, the transmitter receives ACKs more frequently and sends new data earlier when the current flight ends. Another words, the packets from the next flight arrive closer and closer to the end of the first flight. We expect, for a much larger file, the packets will arrive closer and closer until eventually the distinction between flights blurs and the connection settles into a continuous stream of arriving data packets [31]. This is expected to be true provided a large enough advertised window is available. For TCP/IP, if to transmit a 10 Mbytes file, this will not be true since the advertised window starts to limit the sending rate even for a 1000 Kbytes file as we seen in Figure 5.7.

Conclusions

Based on all the above analyses, we may make the following conclusions for the performance comparison of TCP/IP and SCPS over satellite link:

- With channel rate 115,200 bps:115,200 bps, protocols basically have no performance difference.
- With channel rate 4 Mbps:4 Mbps, protocols show "statistically" significant performance differences for large files where SCPS-VJ provides the highest throughput while the performance of SCPS-Vegas is clearly poorer than the other two.
- With channel rate 4 Mbps:57,600 bps, performance tendency is very similar to that for symmetric channel rate 4 Mbps:4 Mbps. This can be shown up from both each protocol has similar performance shape and the performance relationships among all protocols are close. Similar to rate 4 Mbps:4 Mbps, protocols have "statistically" significant performance difference for large files where SCPS-VJ provides the highest throughput and SCPS-Vegas has poor performance.
- For both comparisons with channel rates 4 Mbps:4 Mbps and 4 Mbps:57,600 bps , considering very small mean values for all pairs with even smaller mean differences, their "statistically" significant performance differences are not "practically" significant. Therefore, we may conclude that all protocols have no "really" significant performance differences with channel rates 4 Mbps:4 Mbps and 4 Mbps:57,600 bps in our realistic satellite link experiments. Based

on the above result, we fail to reject **Hypotheses Set 1** and **Hypotheses Set 2** corresponding to the satellite experiment analysis set (1) in Section 3.2.2 and conclude that three protocols have equal file transfer time means for each of the same transmission conditions of BER=0 and Delay=120 ms and three channel rates. But we should note that our conclusion is obtained by not rejecting hypotheses based on the results that all pairs have no "practically" significant performance differences although they have "statistically" significant differences.

5.2 Comparing Protocol Performance over SGLS Test-bed and Satellite Link

In this section, the protocol performance over SGLS test-bed and satellite link is compared to validate the SGLS test-bed performance. The comparison is made only with channel rate 115,200 bps:115,200 bps, BER=0 and Delay=120 ms since this is the only common set experiments conducted in both environments. The comparison for each protocol is done first and then the sample regression lines between two test environments are expected to be built.

Each of two comparison sets (i.e., TCP/IP versus SCPS-VJ and SCPS-Vegas versus SCPS-VJ) has 4 configurations (=2 Test Facility × 2 Protocol) for each file size. The number of observations is 64 (=16 × 4) since there are 16 observations for each configuration.

Protocol TCP/IP

Figure 5.18 plots the protocol TCP/IP performance between the SGLS test-bed and satellite link. The direct observation result is that the performance of two

sources keep spaced for all four files and almost keep parallel for large files. The HSD procedure in Table 5.7 shows that TCP/IP's performance are statistically significant different between the SGLS test-bed and satellite link for all four files.

Protocol SCPS-VJ

The test source performance comparison plot for SCPS-VJ is displayed in Figure 5.19. We see the performance averages bind together for 1K and 10K files and shows big difference for 100K and 1000K files. This is quite different from the source performance plot for TCP/IP. Table 5.8 indicates that the source performance has no difference for 1K and 10K file and has significant difference for 10K and 1000K.

Protocol SCPS-Vegas

Figure 5.20 shows the source performance difference for SCPS-Vegas. We see performance points for test-bed and satellite link are very close for 1K but are widely spaced for large files. Table 5.9 indicates that SCPS-Vegas show significant performance difference for 10K, 100K and 1000K file while has no difference for 1K.

By summarizing the above comparisons for three protocols, we see:

- TCP/IP shows statistically significant difference in the performance for all files between the test-bed and satellite link.
- SCPS-VJ and SCPS-Vegas show statistically significant difference for relatively large files and the difference get large along the increase of the file size.

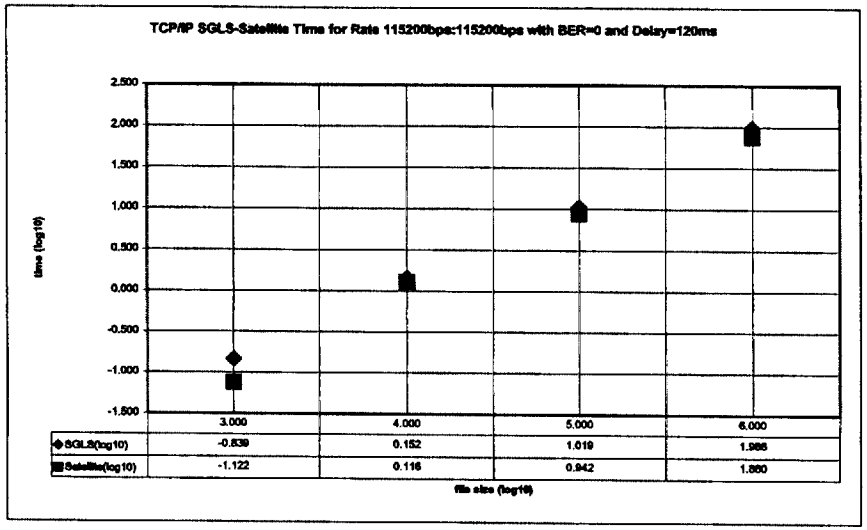

Figure 5.18: Test source comparison for TCP/IP with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

Table 5.5: Comparison of means for test source for TCP/IP with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

File Size	Testbed Mean (secs)	Satellite Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.145	0.075	0.069325	0.053002	0.085648	*	92.4%
10K	1.420	1.307	0.11312	0.05115	0.17510	*	8.7%
100K	10.456	8.750	1.7063	0.7738	2.6387	*	19.5%
1000K	96.875	75.90	20.9750	20.1995	21.7505	*	27.6%

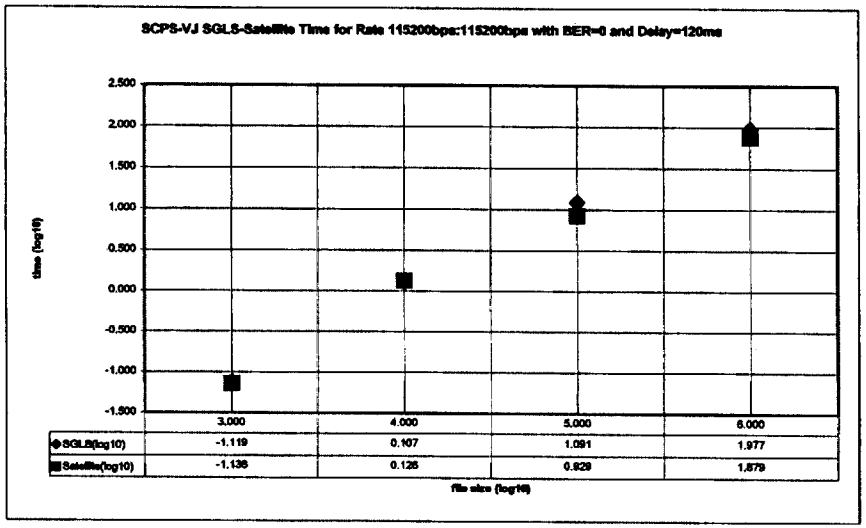

Figure 5.19: Test source comparison for SCPS-VJ with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

Table 5.6: Comparison of means for test source for SCPS-VJ with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

File Size	Testbed Mean (secs)	Satellite Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.076	0.073	0.003538	-0.012786	0.01986		–
10K	1.278	1.337	-0.05907	-0.12104	0.00290		–
100K	12.321	8.486	3.8350	2.9026	4.7674	*	45.2%
1000K	94.823	75.687	19.1360	18.3605	19.9115	*	25.3%

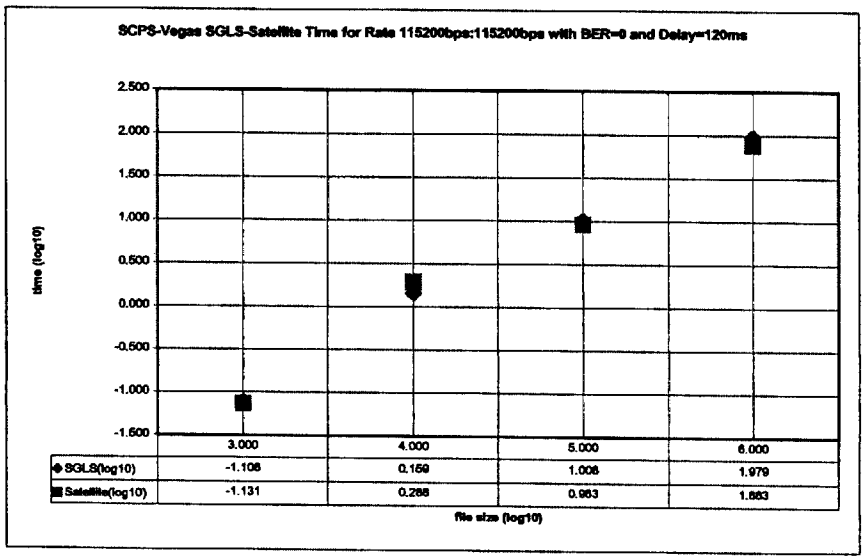

Figure 5.20: Test source comparison for SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

Table 5.7: Comparison of means for test source for SCPS-Vegas with channel rate of 115,200 bps:115,200 bps, BER=0 and Delay=120 ms

File Size	Testbed Mean (secs)	Satellite Mean (secs)	Means Difference (secs)	95% Confidence Limit			Means Difference (%)
1K	0.078	0.074	0.003956	-0.014816	0.02272		–
10K	1.442	1.942	-0.50026	-0.59359	-0.40692	*	-25.7%
100K	10.190	9.180	1.0105	0.0472	1.9738	*	11%
1000K	95.351	76.417	18.9338	17.8427	0.0249	*	24.8%

- All three protocols' performance differences for 1000Kbytes file between the test-bed and satellite link are about 20 seconds that give an approximate 25% mean difference. This indicates that in a steady state, there may be a constant performance bias existing between the test-bed and satellite link. This bias may be related to the difference of testing equipment used for the experiments over test-bed and satellite link. Therefore, we may build a linear regression line for the regression of the SGLS test-bed time on satellite link time. Furthermore, although there may be a constant performance bias existing, the SGLS test-bed may be used to predict the protocol performance over realistic satellite link.

File Transfer Time Relationship between SGLS Test-bed and Satellite

Based on twelve pairs of averaged observations, the linear regression line for the regression of SGLS test-bed time on satellite link time can be found as

$$Satellite\ Time = (0.7923 \times Test\text{-}bed\ Time) + 0.1839 \text{ with } R^2 = 0.9995$$

or in logarithm,

$$\log_{10}(Satellite\ Time) = 0.9888 \times \log_{10}(Test\text{-}bed\ Time) + 0.0605 \text{ with } R^2 = 0.9926$$

This regression line tells us that we can obtain the satellite time if we know the test-bed time. Therefore, we may conclude that our test-bed works well and can be used to predict the protocol performance over realistic satellite link. We also see that, for both models, R^2 is almost 1. This indicates that both models fit the data very well.

Figure 5.21 plots the above model and the file transfer time relationship between SGLS test-bed and satellite link for all three protocols. We see all twelve relationship points are distributed in four clusters separated by four files with each cluster consists of three points corresponding to three protocols. Clearly, we see that SGLS test-bed time and satellite link time are almost linearly correlated each other and increase together. This says that there might be a constant bias existing between them. The regression model verifies this. We see that as we expected the model basically fit all data very well. This constant bias may be related to the difference of testing equipment used for the experiments over test-bed and satellite link.

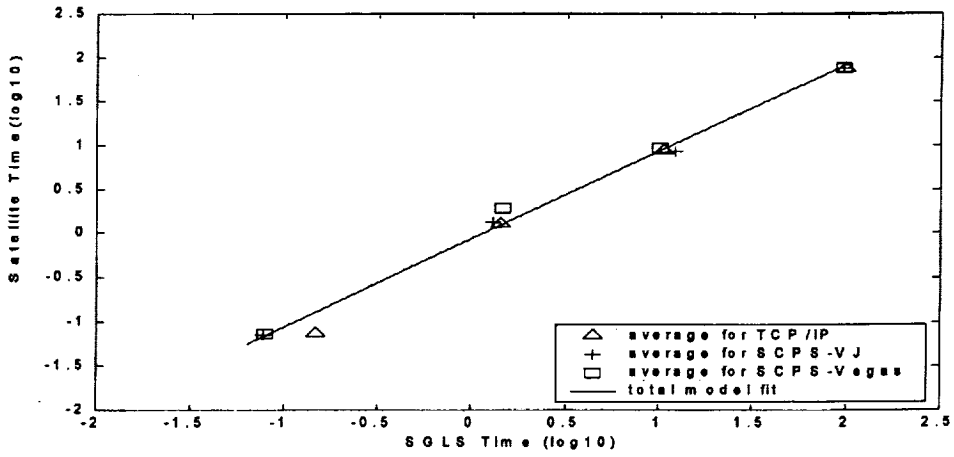

Figure 5.21: File transfer time relationship between SGLS test-bed and satellite link for all three protocols

Considering the different features among three protocols, it may be necessary to find the linear regression line for each protocol. The linear regression line for the regression of test-bed time on satellite link time for each protocol can be found as:

TCP/IP: *Satellite Time* $=(0.7814 \times$ *Test-bed Time*$)+0.2342$ with $R^2=0.9999$

or

$log_{10}(Satellite\ Time) = 1.0535 \times log_{10}(Test\text{-}bed\ Time) - 0.1566$ with $R^2 = 0.9952$

SCPS-VJ: $Satellite\ Time = (0.80 \times Test\text{-}bed\ Time) - 0.3050$ with $R^2 = 0.9995$

or

$log_{10}(Satellite\ Time) = 0.9602 \times log_{10}(Test\text{-}bed\ Time) - 0.0447$ with $R^2 = 0.9977$

SCPS-Vegas: $Satellite\ Time = (0.7960 \times Test\text{-}bed\ Time) + 0.6074$ with $R^2 = 0.9998$

or

$log_{10}(Satellite\ Time) = 0.9665 \times log_{10}(Test\text{-}bed\ Time) + 0.0084$ with $R^2 = 0.9954$

Figures 5.22-5.24 plot the relationship between SGLS test-bed and satellite link and the regression model for TCP/IP, SCPS-VJ and SCPS-Vegas respectively. We see all the above linear models have very high R^2. We expect the models fit data well.

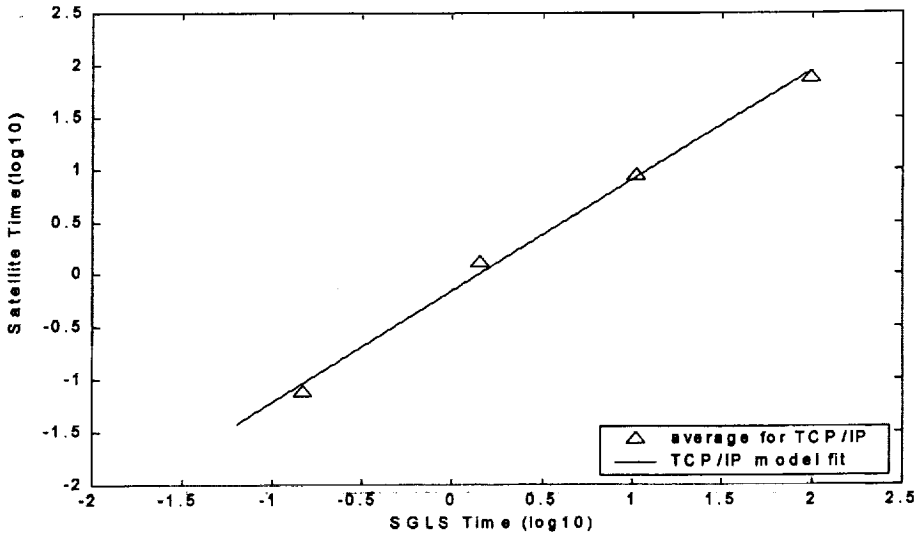

Figure 5.22: TCP/IP file transfer time relationship between SGLS test-bed and satellite link

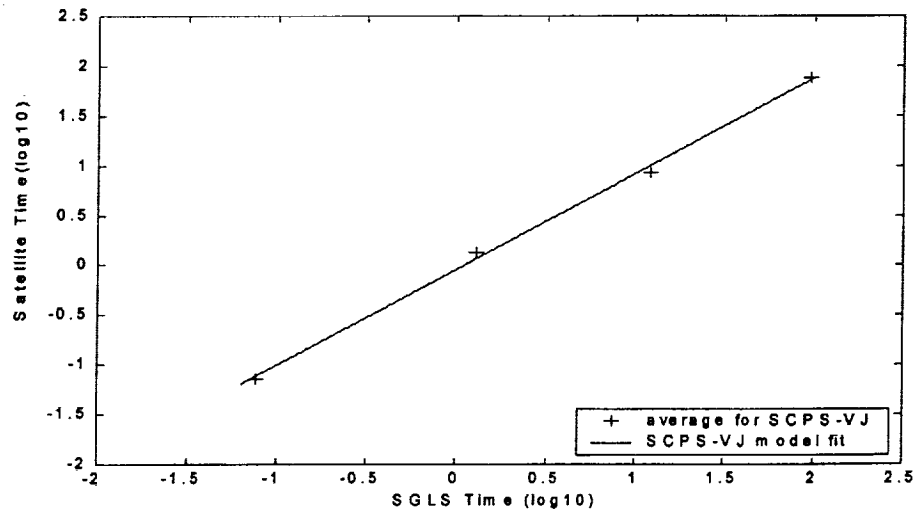

Figure 5.23: SCPS-VJ file transfer time relationship between SGLS test-bed and satellite link

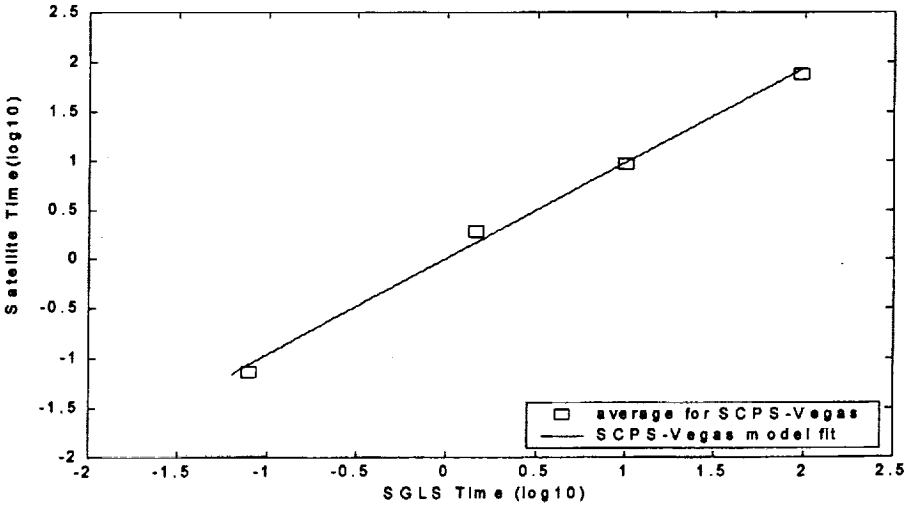

Figure 5.24: SCPS-Vegas file transfer time relationship between SGLS test-bed and satellite link

Conclusions

In summary, based on all the above protocol analyses with BER=0 and Delay=120 ms in Section 5.1 and Sections 5.2, we may conclude:

- All protocols do not show statistically significant mean differences for slow symmetric channel rate 115,200 bps:115,200 bps but show significant performance difference for all large files with higher channel rates.
- SCPS-VJ basically shows the highest throughput in all cases and SCPS-Vegas shows the slowest throughput. All three protocols show statistically significant performance differences between test sources. The mean time for test-bed is about 25% more than that for satellite link for 1000K file for all protocols. Based on this result, we reject **Hypotheses Set 1** through **Hypotheses Set 3** corresponding to the satellite experiment analysis set (2) in Section 3.2.2 and conclude that three protocols have no equal file transfer time means for tests over SGLS test-bed and satellite link with each of four file sizes, channel rate 115,200 bps:115,200 bps, BER=0 and Delay=120 ms. Although existing statistically significant differences in the performance between SGLS test-bed and satellite link, the test-bed works well. There is about 25% constant bias existing between them. This constant bias may be related to the difference of testing equipment used for the experiments over test-bed and satellite link. SGLS test-bed time can be used to predict the protocol performance over satellite link. The prediction may be more accurate when a large file is transmitted.

- The above conclusion addresses our basic question (4): if the SGLS simulator provides a reasonable (to within a scaling factor and offset) approximation to the true satellite channel or if there is a linear translation between the two. The answer is "yes". Based on twelve pairs of averaged observations, the linear regression line for the regression of SGLS test-bed time on satellite link time can be found as

  $$Satellite\ Time = (0.7923 \times Test\text{-}bed\ Time) + 0.1839 \text{ with } R^2 = 0.9995$$

  or in logarithm,

  $$\log_{10}(Satellite\ Time) = 0.9888 \times \log_{10}(Test\text{-}bed\ Time) + 0.0605 \text{ with } R^2 = 0.9926$$

From this, we may conclude that the test-bed works well and can be used to predict the protocol performance over realistic satellite link.

Due to the restriction of satellite link configuration, the experiments with higher BERs and a longer link delay have not been tested. The following work are suggested when satellite link is configured properly: (1) Compare the protocol performance over SGLS test-bed and satellite link for configurations with higher BERs and a longer link delay; (2) Study the effects of BER and link delay on the protocol performance over satellite; (3) Based on a complete protocol performance comparison between TCP and SCPS with various BERs and link delays, the performance relationship between test-bed and satellite link may need revalidation and it may also be necessary to built a new regression model for the regression of test-bed time on satellite time; (4) Study the protocol performance with various BERs and delays over high speed satellite channel rates.

6 SUMMARY, CONCLUSIONS AND FUTURE WORK

We endeavored in this work to study the behavior of TCP and its extensions in space SCPS-TP by testing ftp over TCP/IP stack and SCPS-FP over SCPS-TP/IP stack on both simulated test-bed and realistic satellite link. At the heart of our study lies the protocol performance comparison between TCP/IP suite and SCPS implementations (SCPS-VJ and SCPS-Vegas) and the effects of link delay and BER on each protocol performance.

We wish this study is contributive to understanding most widely used TCP and its extensions in space SCPS, especially on their performance differences. As a joint activity between NASA and DoD, SCPS project is intended to develop data communication protocol standards for data transfer between space mission end systems covering the following technical areas: an efficient file handling protocol (SCPS-FP), various flavors of underlying retransmission control protocol (SCPS-TP), a data protection mechanism (SCPS-SP) and a scaleable network protocol (SCPS-NP). As the developer of both SCPS-TP and SCPS-NP, the MITRE Corporation in Reston, VA also integrated the above four SCPS protocols and conducted a wide range of laboratory tests and live satellite experiments to characterize the performance of SCPS-TP. MITRE also conducted a number of tests in the laboratory to compare the performance of SCPS-TP with that of regular TCP in various simulated space link environments [9]. MITRE's tests were basically concentrated on evaluating the performance of individual protocol segment from the perspective of the protocol developer while our study wished to evaluate the performance of the

complete four-layered-integrated SCPS stack from the perspective of the application user. MITRE's tests were done with each protocol individually and without the operating systems involved and therefore might not be considered as full operations. Our work is considered as the first set of real side-by-side comparison between TCP/IP and SCPS protocol with the operating systems fully involved from the perspective of the user. We wished to determine which protocol suite has better performance under various channel conditions based on real file transmission time reported to the user. It is hoped that our study can provide a reference for application users on the protocol performance difference. We also wished to help protocol developers to better consider the effects of the channel transmission factors on the performance of the integrated complete protocol stack from the perspective of the user, which is actually the work we have been doing.

6.1 Protocols and Experimental Methodologies

The first part of the dissertation introduces TCP and SCPS-TP protocols and describes our experimental facilities, experimental work and analysis procedures. Here we briefly summarize them.

We began by introducing TCP variants and the various congestions control algorithms that dominate the behavior of TCP. Then the communication problems that TCP has in space were listed. Following the problems of TCP, we described how SCPS came out and how TCP was extended to be SCPS-TP to overcome the above TCP problems.

We then introduced NMSU SGLS test-bed and the experimental methodologies. We first described the SGLS test-bed hardware topology and several experimental tools we have and then discussed the protocol layers and configurations in our experiments and the experimental procedures. In the end, we described our detailed experimental work over both test-bed and satellite link and listed testable sets of hypotheses we needed to test.

6.2 Study of Protocols over SGLS Test-bed

The behavior of TCP/IP and SCPS over our test-bed was considered to be one of two essential parts of our study. The goal of this part of our study was to compare the protocol performance over our simulated channel to see which protocol has better performance under different transmission conditions. Another goal was to investigate how each protocol behaves with different delays and BERs and thus to study the effects of delay and BER on their performance. Statistical regression models were also built for the regression of file transfer time on the transmission conditions.

6.2.1 TCP/IP and SCPS Performance Comparison

We compared the protocol performance with different delays and BERs under both symmetric and asymmetric channel rates. We found that:

- Protocols do not show performance difference with a very small file (≤ 1Kbytes) for all configurations.
- For both symmetric and asymmetric channel rates, protocols have no statistically significant performance difference for low BERs with geo-stationary orbit satellite link delay.

- Protocols show statistically significant performance difference with the increase of file size, BER and link delay for both symmetric and asymmetric channel rates. In this case, SCPS-Vegas tends to show the highest throughput and TCP/IP gives the slowest throughput.
- The above conclusion actually answers our first two basic questions listed in Chapter 1. The answer for question (1) is: There is an overall advantage of the SCPS-Vegas protocol for file transport over TCP/IP in our simulated low BDP satellite channel. The answer for question (2) is: Vegas congestion control mode shows superior performance than VJ based congestion control mechanism based on the performance comparison result between SCPS-VJ and SCPS-Vegas.

6.2.2 Delay Effects on Protocol Performance

We analyzed the delay effects on the performance for each of TCP/IP, SCPS-VJ and SCPS-Vegas protocols. We found that:

- Similar to the result for protocol performance comparison, for both symmetric and asymmetric channel rates, all protocols do not show statistically significant performance difference with respect the link delay change for a very small file (≤ 1Kbytes) with all three BERs.
- All protocols show statistically significant different performance with respect to the link delay change along the increase of file size. The difference becomes more significant when the error rate is increased.

- TCP performance difference due to the link delay change along the increase of the file size and BER is larger for asymmetric channel than for symmetric channel; SCPS-VJ and SCPS-Vegas show this difference being stronger for symmetric channel rate. This may be considered as one of key differences between TCP and SCPS implementations.

The above conclusion partially addresses our basic question (3) on how link delay affects each protocol performance.

6.2.3 BER Effects on Protocol Performance

Similar to the study of delay effects, we investigated the BER effects for each of three protocols. We found that:

- When relative small files (< 1Mbytes) were transmitted with symmetric channel rate 115,200 bps:115,200 bps, the increase of BER from error free to 1E-6 does not statistical significantly affect the protocols' performance. BER=1E-5 affects the protocol performance more seriously for all protocols.
- The joint conditions of a large file, a long delay makes all protocols show statistically significant performance difference with respect to the change of BER.
- SCPS-Vegas is sensitive to the increase of BER for transmitting small files over symmetric channel. Reducing the forward channel rate from 115,200 bps to 2400 bps does not significantly affect the SCPS-Vegas performance relationship corresponding to the change of BER if the link delay of 1280 is used.

- The factors of file size, BER and link delay and all their interactions contribute most significantly to the protocol performance.

The above conclusion partially addresses our basic question (3) on how BER affects each protocol performance listed in Chapter 1. When we studied the BER effects on protocol performance, the statistical regression models were also built using BER, link delay and file size as regressors for each protocol over each channel rate.

Based on the above results of protocol study with GEO link delay (120 ms), we may expect that extending it to LEO link (i.e., decreasing link delay to around 3 ms) would not make the protocol performance and the performance relationship too much different since the link delay difference between 3 ms to 120 ms is practically trivial. The verification may be left as part of the future work.

6.3 Study of Protocol Performance over Satellite Link

The study of protocol performance over a real satellite link is another essential part of study. The goals of the study of protocol performance over satellite link are to extend the protocol performance study to cover large BDP region, to compare the protocol performance in practical sense and to validate the SGLS test-bed performance by comparing the in-lab results with the actual satellite channel results under the conditions of slow symmetric channel rate.

Toward the above goals, we first compared the performance of TCP/IP and SCPS over satellite link and then compared protocol performance over SGLS test-bed and satellite link. Three channel rates were available for our experiments over

satellite: (1) 115,200 bps:115,200 bps; (2) 4 Mbps:4 Mbps and (3) 4 Mbps:57,600 bps. The experiments with those rates were done with Delay=120 ms and BER=0.

6.3.1 Comparing Protocol Performance of TCP/IP and SCPS

We compared the performance of TCP/IP and SCPS over satellite link with each of the above three channel rates. For each channel rate, we found that:

- With channel rate 115,200 bps:115,200 bps, protocols basically have no performance difference except for 10K file between SCPS-Vegas and other two protocols.
- With channel rate 4 Mbps:4 Mbps, protocols show statistically significant performance difference for large files where SCPS-VJ provides the highest throughput while the performance of SCPS-Vegas is clearly poorer than other two.
- With channel rate 4 Mbps:57,600 bps, performance tendency is very similar to that for symmetric channel rate 4 Mbps:4 Mbps. This can be shown up from both each protocol has similar performance shape and the performance relationships among all protocols are close. Similar to rate 4 Mbps:4 Mbps, protocols have statistically significant performance difference for large files where SCPS-VJ provides the highest throughput and SCPS-Vegas has poor performance.

Combining the above conclusions for tests over satellite link, we concluded:

- All protocols do not show statistically significant performance differences for slow symmetric channel rate 115,200 bps:115,200 bps but show significant performance difference for all large files with higher channel rates.
- SCPS-VJ basically shows the highest throughput in all cases and SCPS-Vegas shows the slowest throughput.
- For both comparisons with channel rates 4 Mbps:4 Mbps and 4 Mbps:57,600 bps, considering very small mean values for all pairs with even smaller mean differences, their "statistically" significant performance differences are not "practically" significant. Therefore, we may conclude that all protocols have no "really" significant performance differences with channel rates 4 Mbps:4 Mbps and 4 Mbps:57,600 bps in our realistic satellite link experiments.

To see how three protocols perform differently in detail, we also analyzed the protocol behavior for transmitting 1000 Kbytes file with rate 4 Mpbs:57,600 bps, BER=0 and Delay=120 ms using various network performance graphs.

6.3.2 Comparing Protocol Performance over SGLS Test-bed and Satellite Link

The protocol performance over SGLS test-bed and satellite link is compared to validate the SGLS test-bed performance. The comparison is made only with the channel rate 115,200 bps:115,200 bps, BER=0 and Delay=120 ms since this is the only common set experiments conducted in both test-bed and satellite link. We found that:

- TCP/IP shows statistically significant difference in the performance for all files between the test-bed and satellite link.

- SCPS-VJ and SCPS-Vegas show statistically significant difference for relatively large files and the difference get large along the increase of the file size.
- All three protocols' performance differences for 1000Kbytes file between the test-bed and satellite link are about 20 seconds that give an approximate 25% mean difference. This indicates that in a steady state, there may be a constant performance bias existing between the test-bed and satellite link. This bias may be related to the difference of testing equipment used for the experiments over test-bed and satellite link

The overall protocol performance over test-bed and satellite link is linearly correlated by

$$Satellite\ Time = (0.7923 \times Test\text{-}bed\ Time) + 0.1839 \text{ with } R^2 = 0.9995$$

or in logarithm,

$$\log_{10}(Satellite\ Time) = 0.9888 \times \log_{10}(Test\text{-}bed\ Time) + 0.0605 \text{ with } R^2 = 0.9926$$

This addresses our basic question (4) in Chapter 1. The constant bias may be related to the difference of testing equipment used for the experiments over test-bed and satellite link. We concluded that the test-bed works well and can be used to predict the protocol performance over realistic satellite link.

6.4 Future Work

Due to the restriction of satellite link configuration, the experiments with higher BERs and a longer link delay have not been tested. The following work are suggested when satellite link is configured properly:

- Compare the protocol performance over SGLS test-bed and satellite link for configurations with higher BERs and a longer link delay.
- Study the effects of BER and link delay on the protocol performance over various satellite channel rates.
- Based on a complete protocol performance comparison between TCP and SCPS with various BERs and link delays, the performance relationship between test-bed and satellite link will need to be reviewed and it will be necessary to built a new regression model for the regression of test-bed time on satellite time. It might also be necessary to re-validate test-bed performance based on additional experiments and this new regression model.

APPENDICES

A. MEANS AND STANDARD DEVIATIONS OF TCP/IP FILE TIMES OVER THE SGLS TEST-BED

Protocol	Channel Rate	BER	File Size (bytes)	Delay (ms)	Runs	Transfer Time Means (secs)	Transfer Time Std Dev (secs)
TCP/IP	SYM	0	997	120	16	0.145	0.017
TCP/IP	SYM	0	997	1280	16	0.137	0.019
TCP/IP	SYM	0	10007	120	16	1.420	0.067
TCP/IP	SYM	0	10007	1280	16	3.634	0.124
TCP/IP	SYM	0	100003	120	16	10.456	0.171
TCP/IP	SYM	0	100003	1280	16	20.244	1.690
TCP/IP	SYM	0	1000003	120	16	96.875	0.649
TCP/IP	SYM	0	1000003	1280	16	120.438	13.366
TCP/IP	SYM	1.00E-06	997	120	16	0.126	0.024
TCP/IP	SYM	1.00E-06	997	1280	16	0.147	0.021
TCP/IP	SYM	1.00E-06	10007	120	16	1.314	0.170
TCP/IP	SYM	1.00E-06	10007	1280	16	3.697	0.149
TCP/IP	SYM	1.00E-06	100003	120	16	10.544	0.876
TCP/IP	SYM	1.00E-06	100003	1280	16	27.669	8.507
TCP/IP	SYM	1.00E-06	1000003	120	16	100.519	4.238
TCP/IP	SYM	1.00E-06	1000003	1280	16	239.125	15.379
TCP/IP	SYM	0.00001	997	120	16	0.312	0.750
TCP/IP	SYM	0.00001	997	1280	16	0.328	0.742
TCP/IP	SYM	0.00001	10007	120	16	2.450	2.506
TCP/IP	SYM	0.00001	10007	1280	16	8.097	3.474
TCP/IP	SYM	0.00001	100003	120	16	15.088	3.220
TCP/IP	SYM	0.00001	100003	1280	16	66.919	15.870
TCP/IP	SYM	0.00001	1000003	120	16	146.063	13.453
TCP/IP	SYM	0.00001	1000003	1280	16	717.688	34.358
TCP/IP	USYM	0	997	120	16	0.145	0.060
TCP/IP	USYM	0	997	1280	16	0.140	0.017
TCP/IP	USYM	0	10007	120	16	2.128	0.123
TCP/IP	USYM	0	10007	1280	16	8.689	4.217
TCP/IP	USYM	0	100003	120	16	17.269	2.084
TCP/IP	USYM	0	100003	1280	16	50.156	1.332
TCP/IP	USYM	0	1000003	120	16	173.500	3.464
TCP/IP	USYM	0	1000003	1280	16	222.438	17.029
TCP/IP	USYM	1.00E-06	997	120	16	0.154	0.062
TCP/IP	USYM	1.00E-06	997	1280	16	0.138	0.014
TCP/IP	USYM	1.00E-06	10007	120	16	2.338	0.524
TCP/IP	USYM	1.00E-06	10007	1280	16	7.657	0.065
TCP/IP	USYM	1.00E-06	100003	120	16	20.025	2.286

Protocol	Channel Rate	BER	File Size (bytes)	Delay (ms)	Runs	Transfer Time Mean (secs)	Transfer Time Std Dev (secs)
TCP/IP	USYM	1.00E-06	100003	1280	16	57.744	7.400
TCP/IP	USYM	1.00E-06	1000003	120	16	194.625	2.553
TCP/IP	USYM	1.00E-06	1000003	1280	16	355.813	21.389
TCP/IP	USYM	0.00001	997	120	16	0.521	1.005
TCP/IP	USYM	0.00001	997	1280	16	0.515	1.487
TCP/IP	USYM	0.00001	10007	120	16	3.700	1.577
TCP/IP	USYM	0.00001	10007	1280	16	20.935	20.968
TCP/IP	USYM	0.00001	100003	120	16	30.925	2.596
TCP/IP	USYM	0.00001	100003	1280	16	83.744	11.778
TCP/IP	USYM	0.00001	1000003	120	16	302.563	12.801
TCP/IP	USYM	0.00001	1000003	1280	16	998.438	28.847

Channel Rate:

SYM —115,200 bps:115,200 bps

USYM —115,200 bps:2400 bps

B. MEANS AND STANDARD DEVIATIONS OF SCPS-VJ FILE TIMES OVER THE SGLS TEST-BED

Protocol	Channel Rate	BER	File Size (bytes)	Delay (ms)	Runs	Transfer Time Means (secs)	Transfer Time Std Dev (secs)
SCPS/VJ	SYM	0	997	120	16	0.076	0.028
SCPS/VJ	SYM	0	997	1280	16	0.082	0.031
SCPS/VJ	SYM	0	10007	120	16	1.278	0.113
SCPS/VJ	SYM	0	10007	1280	16	3.612	0.113
SCPS/VJ	SYM	0	100003	120	16	12.321	1.988
SCPS/VJ	SYM	0	100003	1280	16	18.729	0.480
SCPS/VJ	SYM	0	1000003	120	16	94.823	1.528
SCPS/VJ	SYM	0	1000003	1280	16	137.043	7.501
SCPS/VJ	SYM	1.00E-06	997	120	16	0.082	0.047
SCPS/VJ	SYM	1.00E-06	997	1280	16	0.082	0.027
SCPS/VJ	SYM	1.00E-06	10007	120	16	1.219	0.127
SCPS/VJ	SYM	1.00E-06	10007	1280	16	3.829	0.609
SCPS/VJ	SYM	1.00E-06	100003	120	16	11.585	1.883
SCPS/VJ	SYM	1.00E-06	100003	1280	16	23.929	4.223
SCPS/VJ	SYM	1.00E-06	1000003	120	16	97.910	2.436
SCPS/VJ	SYM	1.00E-06	1000003	1280	16	207.292	10.429
SCPS/VJ	SYM	0.00001	997	120	16	0.167	0.241
SCPS/VJ	SYM	0.00001	997	1280	16	0.435	0.943
SCPS/VJ	SYM	0.00001	10007	120	16	1.484	0.249
SCPS/VJ	SYM	0.00001	10007	1280	16	5.340	1.768
SCPS/VJ	SYM	0.00001	100003	120	16	13.023	1.514
SCPS/VJ	SYM	0.00001	100003	1280	16	51.649	4.716
SCPS/VJ	SYM	0.00001	1000003	120	16	139.901	9.278
SCPS/VJ	SYM	0.00001	1000003	1280	16	567.325	12.581
SCPS/VJ	USYM	0	997	120	16	0.089	0.025
SCPS/VJ	USYM	0	997	1280	16	0.086	0.029
SCPS/VJ	USYM	0	10007	120	16	2.255	0.120
SCPS/VJ	USYM	0	10007	1280	16	4.231	0.082
SCPS/VJ	USYM	0	100003	120	16	17.022	0.381
SCPS/VJ	USYM	0	100003	1280	16	24.205	1.878
SCPS/VJ	USYM	0	1000003	120	16	174.482	1.843
SCPS/VJ	USYM	0	1000003	1280	16	190.625	5.874
SCPS/VJ	USYM	1.00E-06	997	120	16	0.088	0.015
SCPS/VJ	USYM	1.00E-06	997	1280	16	0.081	0.019
SCPS/VJ	USYM	1.00E-06	10007	120	16	2.350	0.415
SCPS/VJ	USYM	1.00E-06	10007	1280	16	4.240	0.111
SCPS/VJ	USYM	1.00E-06	100003	120	16	19.132	1.880
SCPS/VJ	USYM	1.00E-06	100003	1280	16	29.338	5.949

Protocol	Channel Rate	BER	File Size (bytes)	Delay (ms)	Runs	Transfer Time Mean (secs)	Transfer Time Std Dev (secs)
SCPS/VJ	USYM	1.00E-06	1000003	120	16	192.331	2.018
SCPS/VJ	USYM	1.00E-06	1000003	1280	16	250.517	6.630
SCPS/VJ	USYM	0.00001	997	120	16	0.158	0.258
SCPS/VJ	USYM	0.00001	997	1280	16	0.506	1.174
SCPS/VJ	USYM	0.00001	10007	120	16	2.840	0.886
SCPS/VJ	USYM	0.00001	10007	1280	16	6.046	1.619
SCPS/VJ	USYM	0.00001	100003	120	16	27.596	1.971
SCPS/VJ	USYM	0.00001	100003	1280	16	65.687	8.209
SCPS/VJ	USYM	0.00001	1000003	120	16	274.758	4.998
SCPS/VJ	USYM	0.00001	1000003	1280	16	669.597	9.756

Channel Rate:

SYM —115,200 bps:115,200 bps

USYM —115,200 bps:2400 bps

C. MEANS AND STANDARD DEVIATIONS OF SCPS-VEGAS FILE TIMES OVER THE SGLS TEST-BED

Protocol	Channel Rate	BER	File Size (bytes)	Delay (ms)	Runs	Transfer Time Means (secs)	Transfer Time Std Dev (secs)
SCPS/VG	SYM	0	997	120	16	0.078	0.027
SCPS/VG	SYM	0	997	1280	16	0.073	0.016
SCPS/VG	SYM	0	10007	120	16	1.442	0.163
SCPS/VG	SYM	0	10007	1280	16	6.074	0.152
SCPS/VG	SYM	0	100003	120	16	10.190	0.546
SCPS/VG	SYM	0	100003	1280	16	22.675	1.515
SCPS/VG	SYM	0	1000003	120	16	95.351	1.755
SCPS/VG	SYM	0	1000003	1280	16	135.183	4.938
SCPS/VG	SYM	1.00E-06	997	120	16	0.078	0.013
SCPS/VG	SYM	1.00E-06	997	1280	16	0.084	0.024
SCPS/VG	SYM	1.00E-06	10007	120	16	1.568	0.071
SCPS/VG	SYM	1.00E-06	10007	1280	16	6.086	0.109
SCPS/VG	SYM	1.00E-06	100003	120	16	11.052	1.156
SCPS/VG	SYM	1.00E-06	100003	1280	16	24.102	1.546
SCPS/VG	SYM	1.00E-06	1000003	120	16	97.402	0.493
SCPS/VG	SYM	1.00E-06	1000003	1280	16	162.009	4.692
SCPS/VG	SYM	0.00001	997	120	16	0.109	0.102
SCPS/VG	SYM	0.00001	997	1280	16	0.603	1.119
SCPS/VG	SYM	0.00001	10007	120	16	1.789	0.269
SCPS/VG	SYM	0.00001	10007	1280	16	7.351	2.240
SCPS/VG	SYM	0.00001	100003	120	16	14.006	4.317
SCPS/VG	SYM	0.00001	100003	1280	16	33.944	3.487
SCPS/VG	SYM	0.00001	1000003	120	16	117.240	3.682
SCPS/VG	SYM	0.00001	1000003	1280	16	267.870	12.830
SCPS/VG	USYM	0	997	120	16	0.095	0.016
SCPS/VG	USYM	0	997	1280	16	0.078	0.022
SCPS/VG	USYM	0	10007	120	16	2.814	0.140
SCPS/VG	USYM	0	10007	1280	16	7.199	0.072
SCPS/VG	USYM	0	100003	120	16	18.976	0.593
SCPS/VG	USYM	0	100003	1280	16	27.754	0.662
SCPS/VG	USYM	0	1000003	120	16	177.758	4.026
SCPS/VG	USYM	0	1000003	1280	16	193.112	4.967
SCPS/VG	USYM	1.00E-06	997	120	16	0.159	0.291
SCPS/VG	USYM	1.00E-06	997	1280	16	0.068	0.016
SCPS/VG	USYM	1.00E-06	10007	120	16	2.867	0.222
SCPS/VG	USYM	1.00E-06	10007	1280	16	7.245	0.182
SCPS/VG	USYM	1.00E-06	100003	120	16	20.027	1.279
SCPS/VG	USYM	1.00E-06	100003	1280	16	29.958	2.119
SCPS/VG	USYM	1.00E-06	1000003	120	16	189.916	3.984

Protocol	Channel Rate	BER	File Size (bytes)	Delay (ms)	Runs	Transfer Time Mean (secs)	Transfer Time Std Dev (secs)
SCPS/VG	USYM	1.00E-06	1000003	1280	16	223.997	5.509
SCPS/VG	USYM	0.00001	997	120	16	0.396	1.218
SCPS/VG	USYM	0.00001	997	1280	16	0.495	1.196
SCPS/VG	USYM	0.00001	10007	120	16	3.422	0.960
SCPS/VG	USYM	0.00001	10007	1280	16	8.872	2.070
SCPS/VG	USYM	0.00001	100003	120	16	28.595	1.550
SCPS/VG	USYM	0.00001	100003	1280	16	40.146	2.090
SCPS/VG	USYM	0.00001	1000003	120	16	261.204	5.235
SCPS/VG	USYM	0.00001	1000003	1280	16	348.515	13.598

SCPS/VG — Protocol SCPS-Vegas

Channel Rate:

SYM —115,200 bps:115,200 bps

USYM —115,200 bps:2400 bps

D. MEANS AND STANDARD DEVIATIONS OF PROTOCOL FILE TIMES OVER SATELLIE LINK

Protocol	Channel Rate	BER	File Size (bytes)	Delay (ms)	Runs	Transfer Time Mean (secs)	Transfer Time Std Dev (secs)
TCP/IP	1152	0	997	120	16	0.075	0.011
TCP/IP	1152	0	10007	120	16	1.307	0.013
TCP/IP	1152	0	100003	120	16	8.750	0.016
TCP/IP	1152	0	1000003	120	16	75.900	0.000
TCP/IP	4Mbps	0	997	120	16	0.005	0.000
TCP/IP	4Mbps	0	10007	120	16	0.555	0.001
TCP/IP	4Mbps	0	100003	120	16	3.282	0.009
TCP/IP	4Mbps	0	1000003	120	16	6.591	0.024
TCP/IP	576	0	997	120	16	0.003	0.003
TCP/IP	576	0	10007	120	16	0.718	0.008
TCP/IP	576	0	100003	120	16	4.246	0.005
TCP/IP	576	0	1000003	120	16	8.474	0.030
SCPS/VJ	1152	0	997	120	16	0.073	0.006
SCPS/VJ	1152	0	10007	120	16	1.337	0.011
SCPS/VJ	1152	0	100003	120	16	8.486	0.024
SCPS/VJ	1152	0	1000003	120	16	75.687	0.017
SCPS/VJ	4Mbps	0	997	120	16	0.003	0.001
SCPS/VJ	4Mbps	0	10007	120	16	0.741	0.011
SCPS/VJ	4Mbps	0	100003	120	16	3.066	0.086
SCPS/VJ	4Mbps	0	1000003	120	16	6.386	0.091
SCPS/VJ	576	0	997	120	16	0.002	0.001
SCPS/VJ	576	0	10007	120	16	0.913	0.003
SCPS/VJ	576	0	100003	120	16	3.766	0.004
SCPS/VJ	576	0	1000003	120	16	7.892	0.054
SCPS/VG	1152	0	997	120	16	0.074	0.008
SCPS/VG	1152	0	10007	120	16	1.942	0.015
SCPS/VG	1152	0	100003	120	16	9.180	0.014
SCPS/VG	1152	0	1000003	120	16	76.417	0.197
SCPS/VG	4Mbps	0	997	120	16	0.004	0.000
SCPS/VG	4Mbps	0	10007	120	16	1.093	0.000
SCPS/VG	4Mbps	0	100003	120	16	3.833	0.006
SCPS/VG	4Mbps	0	1000003	120	16	7.491	0.003
SCPS/VG	576	0	997	120	16	0.003	0.002
SCPS/VG	576	0	10007	120	16	1.402	0.004
SCPS/VG	576	0	100003	120	16	4.915	0.003
SCPS/VG	576	0	1000003	120	16	9.631	0.007

SCPS/VG — Protocol SCPS-Vegas

Channel Rate:

1152 —115,200 bps:115,200 bps

4 Mbps — 4 Mbps:4 Mbps

576 — 4 Mbps:57,600 bps

REFERENCES

[1] Mark Allman, Chris Heyes, Hans Kruse, Shawn Osterman, "TCP Performance over Satellite Links," *Proc. 5th International Conf. on Telecom. Systems*, 1997.

[2] Yongguang Zhang, Elliot Yan, Son K. Dao, "A Measurement of TCP over Long-Delay Network," *Proc. 6th International Conf. on Telecom. Systems*, March 1998.

[3] M. Mathis, J. Mahdavi, PSC, S. Floyd, "TCP Selective Acknowledgment Options," IETF, Read for Comments RFC 2018, October 1996.

[4] Consultative Committee for Space Data Systems, "Space Communications Protocols Specification (SCPS)-Rational, Requirements, and Application Notes," CCSDS 710.0-G-0.3, April 1997.

[5] S. Horan and R. H. Wang, "Design of a Channel Error Simulator Using Virtual Instrument Techniques for the Initial Testing of TCP/IP and SCPS Protocols," NMSU-ECE-99-002, April 1999.

[6] S. Horan and R. Wang, "Effects of Networking Configurations on Channel Throughput in Space," *Proc. 14th AIAA/USU Conference on Small Satellites*, August 2000.

[7] S. Horan and R. Wang, "Internet-Type Protocol Testing in a Simulated Small Satellite Environment," *IEEE Conference on Aerospace and Electronic Systems*, March 2001.

[8] S. Horan and R. H. Wang, "Effects of Protocol Options on Data Throughput over a Simulated Space Channel," NMSU-ECE-00-007, August 2000.

[9] R. Durst, G. Miller and E. Travis, "TCP Extensions for Space Communications," *Proc. of the 2nd ACM Conference on Mobile Computing and Network*, November 1996.

[10] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithm," IETF, Request for Comments, RFC 2001, January 1997.

[11] V. Jacobson, "Congestion Avoidance and Control," *Proc. SIGCOMM'88*, August 1988, pp. 314-329.

[12] L. Brakmo and L. Peterson, "TCP Vegas: End-to-End Congestion Avoidance on a Global Internet," *IEEE Journal on Selected Area in Communications*, vol. 13,

pp. 1465-1480. October 1995.

[13] S. Floyd and T. Henderson, "The New Reno modification to TCP's fast recovery algorithm," IETF Internet Draft, draft-ietf-tcpimpl-newreno-00.txt

[14] J. R. Chen and Y. C. Chen, "Vegas plus: Improving the service fairness," *IEEE Communication Letters*, vol. 4, no. 5, May 2000.

[15] M. Mathis and J. Mahdavi, "Forward acknowledgment: Refining TCP congestion control," *Computer Communication Review*, vol. 26, no. 4, pp. 281-291, 1996.

[16] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, Reading, MA, 1994.

[17] K. Fall and S. Floyd, "Simulation-based comparison of Tahoe, Reno, and SACK TCP," *Computer Communication Review*, vol. 26, no. 3, July 1996.

[18] S. Floyd. "Issues of TCP with SACK," Technical report, March 1996 URL <*ftp://ftp.ee.lbl.gov/papers/issues_sa.ps.Z*>.

[19] U. Hengartner, J. Bolliger and Th. Gross, "TCP Vegas revisited," *INFOCOM'00.*

[20] L. Brakmo, S. O'Malley and L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," *Proc. of ACM SIGCOMM'94*, pp. 24-35. October 1994.

[21] C. Partridge, *Gigabit Networking*, Addison-Wesley, Reading, MA, 1994.

[22] V. Jacobson, "Compressing TCP/IP Header for Low-Speed Serial Links," IETF Request for Comments RFC1144, February 1990.

[23] R. Fox, "TCP big window and Nak Options," IETF Request for Comments RFC 1106, June 1989.

[24] V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance," IETF Request for Comments RFC 1323, May 1992

[25] V. Jacobson and R. Braden, "TCP Extensions for Long-Delay Paths," IETF Request for Comments RFC 1072, October 1988.

[26] S. Dowdy and S. Wearden, *Statistics for Research*, 2nd Edition, John Wiley & Sons, New York, 1991.

[27] SAS Institute Inc., *SAS User's Guide: Statistics, Version 5 Edition*, SAS Circle,

Cary, NC, 1985.

[28] D. A. Belsley, E. Kuh and R. E. Welsch, *Regression Diagnostics: Identifying Influential Data and Source of Collinearity*, New York: Wiley, 1980.

[29] R. D. Cook and S. Weisberg, *Residuals and Influence in Regression*, London: Chapman and Hall, 1982.

[30] N. R. Draper and H. Smith, *Applied Regression Analysis*, 2nd Edition, New York: Wiley, 1981.

[31] Vern Paxson, "Automated Packet Trace Analysis of TCP Implementations," *Proc. SIGCOMM'97.*